DOCUMENT RESUME

ED 274 516 SE 047 206

AUTHOR	Vernier, David L.
TITLE	How to Build a Better Mousetrap and 13 Other Science Projects Using the Apple II.
REPORT NO	ISBN-0-918731-16-X
PUB DATE	86
NOTE	225p.
AVAILABLE FROM	Vernier Software, 2920 S.W. 89th Ave., Portland, OR 97225 ($24.95, includes sample programs disk and shipping).
PUB TYPE	Guides - Classroom Use - Guides (For Teachers) (052) -- Computer Programs (101)
EDRS PRICE	MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS	*Computer Assisted Instruction; Computer Uses in Education; Courseware; Laboratory Equipment; Laboratory Experiments; Microcomputers; *Physical Sciences; *Science Activities; Science Education; *Science Experiments; Science Instruction; Science Projects; Secondary Education; *Secondary School Science; Technological Advancement
IDENTIFIERS	*Apple II; *Laboratory Interfacing

ABSTRACT

Science projects which can be used by high school students who have minimal experience with computers or electronics are presented in this book on laboratory interfacing. These laboratory interfacing projects include either the connecting of measuring instruments directly to a computer or using a computer to control external devices. All of the 14 projects of this book involve construction of electric circuits that connect to the Apple II plus or IIe game port. In each case, a core project is discussed and step-by-step instructions are given for its assembly and testing. Descriptions of each project include: (1) project requirements (containing electronics and programming information, lists of parts, supplies, and tools); (2) background information (examining the scientific basis of the project); (3) step-by-step instructions (specifying the methodology in building and using the project); (4) troubleshooting (offering corrective suggestions); (5) project extensions (recommending additional experiments); and (6) supplemental information (consisting of science and mathematics information and data on electronic components). A disk with sample programs is available. (ML)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

HOW TO BUILD A BETTER MOUSETRAP

and 13 Other Science Projects Using the Apple II®

Fourteen educational projects for making measurements and controlling electrical devices through the Apple II+ or IIe game port. Includes a disk of sample programs.

How to Build a Better Mousetrap and 13 Other Science Projects Using the Apple II. ® Copyright © 1986 by David L. Vernier. All rights reserved. No part of this book may be used or reproduced in any manner whatsoever without written permission of the publisher except in the case of brief quotations embodied in critical articles or reviews.

Every effort has been made to supply complete and accurate information. However, Vernier Software assumes no responsibility for its use, nor for any infringements of patents or other rights of third parties that would result. Neither is any liability assumed for damages resulting from the use of the information contained herein.

The terms Apple, Applesoft, Apple II, Apple II+, Apple IIe, Apple IIc, LaserWriter, DOS 3.3, ProDOS, AppleWorks, FILER and CONVERT are either registered trademarks, trademarks, or copyrighted by Apple Computer, Inc.

Published by
Vernier Software
2920 S.W. 89th Avenue
Portland, OR 97225
(503) 297-5317

Photographs by Sheila M. Cronin

Library of Congress Card Number: 86-090308
ISBN 0-918731-16-X
First Edition 1986
First Printing
Printed in the United States of America

To my father, who encouraged me to be curious and skeptical.

Acknowledgments

Many people helped in making this book a reality. Special thanks to all of the following:

> Our many editors and test sites improved the writing of a physics major and endured many soldering-iron burns. They made many suggestions that have improved this book: Brad Huff, Patricia Vernier, Janet L. White, Barbara Parish Karn, William Lamb, Robert B. Teese, Walter Hellman, Marcia Stevenson, John Neumann, Robert Pease and John Stotka and Peter Harr.
>
> Robert Cary of Sunset Laboratory, Matt Zimmerman of ZYX Measurements and Dr. David Lineback of Measurements Group, Inc. provided invaluable advice on electronic circuits.
>
> Tom Wesselman built and critiqued every project and program.
>
> Sheila M. Cronin produced all of the photographs of the projects.
>
> Apple Computer, Inc. manufactured the Macintosh™ computer and the LaserWriter™ printer that make this kind of desktop publishing possible.

Thanks, also, to all of the customers of Vernier Software who have encouraged us in doing this project.

Last, but not least, I thank my wife Christine Vernier, who should actually be listed as the co-author. She has spent many long hours assembling, testing, reading and rewriting each project. Even though she did not know a capacitor from a resistor when she started, she got all her projects to work. I am sure she has learned more about electronics and science than she ever wanted to know.

Contents

Introduction xii

- The *Project Programs* Disk xv
- Electronic Components xvi
- Sources of Electronic Components xxii
- Project Assembly Tips xxiv
- The Apple Game Port xxvii

Project 1. Photogate Timer 1-1
Project 2. Reaction Time 2-1
Project 3. Microphone/Amplifier 3-1
Project 4. Humidity Meter 4-1
Project 5. Resistance/Capacitance Meter 5-1
Project 6. IC Temperature Probe 6-1
Project 7. Voltage Monitor 7-1
Project 8. Thermocouple 8-1
Project 9. pH Meter 9-1
Project 10. Strain Gage 10-1
Project 11. Optically-Isolated Switches 11-1
Project 12. Computer-Controlled Car 12-1
Project 13. Stepper Motor 13-1
Project 14. A Better Mousetrap 14-1

Appendixes

A. Cables to the Apple Game Ports A-1
B. Resistor Color Codes B-1
C. Bibliography C-1
D. Software on the *Project Programs* Disk D-1
E. Assembly- and Machine-Language Program Listings E-1
F. Converting the Programs to ProDOS F-1
G. Teacher's Guide G-1
H. Other Vernier Software Products H-1

Parts Kit Order Form

Feedback Form

INTRODUCTION

One of the most popular and exciting ways to use a computer these days is for *laboratory interfacing*. Laboratory interfacing usually refers to connecting measurement instruments directly to a computer. This allows data to be collected and displayed automatically. The possibilities are unlimited. Computers can be used to measure temperature, humidity, force, pH, frequency, time intervals, etc. Laboratory interfacing can also refer to using a computer to control external devices, such as motors, lights, heaters and robots. The fourteen projects in this book include both types of laboratory interfacing.

Laboratory interfacing is traditionally done using special interfacing boards that are plugged into slots inside the computer. These boards include circuits that help the computer sense or control the "outside world." Interfacing boards usually cost several hundred dollars. They are available for almost any type of computer, including the Apple II+ and IIe. Fortunately, a lot of laboratory interfacing can be done with the Apple II+ or IIe computer through its game port.[1] This saves the cost of the interfacing board. All of the projects in this book interface through the game port.

There is no better way to learn about laboratory interfacing than by "jumping into it." Using the Apple II+ or IIe, you can build laboratory interfacing projects inexpensively and with very little chance of damaging your computer. At the same time, you can learn some science, electronics and programming. As a fringe benefit, you probably will have fun!

This book can help you learn about laboratory interfacing. All of the projects involve construction of electronic circuits that connect to the Apple II+ or IIe game port and allow the computer to interact with the outside world. The projects are designed to be done by students with minimal experience in electronics and computers. In each case, a core project is discussed and step-by-step instructions are given for its assembling and testing. After the core project is completed, suggestions are offered for project extensions. Expanding the project can include improving the electronic circuit, adding new features to the programs that support the project or doing additional experiments.[2]

Each project is described in a chapter that consists of:

- **Project Requirements** - Information on the background knowledge and experience needed in electronics and programming, lists of parts, supplies and tools needed for the project.
- **Background Information** - A look at the electronics and the science behind the project.
- **Step-by-Step Instructions** - Detailed instructions on how to build and use the core project.
- **Troubleshooting** - Suggestions on what to do if the circuit does not work correctly.
- **Project Extensions** - Suggestions for extending the project by changing the hardware, improving the software, or doing additional experiments with the device.
- **Supplemental Information** - Additional science and math background information, data on the electronic components used, or explanations of how the programs operate.

---

[1] The projects in this book are not designed for the Apple IIc computer. The game/mouse port on the IIc is not the same electrically as the game port on the II+ and IIe. Projects 2, 5, and 6 will work on the IIc with no changes. Most of the other projects can be made to work by slightly modifying the circuits. Projects 11-14 are impossible on the IIc.

[2] We would like to hear about interesting project extensions that you develop. We may mention your ideas in future revisions of this book. A *Feedback Form* is provided in the back of the book.

The *Project Programs* disk included with this book has sample programs that will let you try out your project with the computer. Most of these are short, straightforward BASIC programs. They include many comments (in REM statements) to explain how the programs operates.

Before you begin assembling anything, we recommend you read the entire *Introduction* of this book, which contains information that will be helpful in selecting parts and in assembling the projects. It also gives you an explanation of the various connections provided on the Apple II+ and IIe game ports.

When selecting a project to undertake, look over the "Project Requirements" at the beginning of each chapter. Choose a project that is within your abilities. Some projects require previous electronics-assembly experience; a few require knowledge of BASIC programming. Do not "bite off more than you can chew." Remember, you can always expand on a project if time allows. In fact, the core project explained in the "Step-by-Step Instructions" may be just the starting point for an impressive science project. The chart on the next page may also be used to help you decide on the right project. In the first column, each project has been assigned an arbitrary "difficulty factor." This number is based on discussions with our test assemblers. If you have never soldered together an electronic circuit, start with one of the projects that has a low difficulty factor. After you do a few of these, you will be ready for a more complex project. The chart also indicates the approximate time that our test assemblers spent building each core project. The time listed does not include time spent doing project extensions.

Another factor to consider in selecting a project is the availability and cost of the necessary parts. Column four of the chart indicates whether or not it is likely that you will be able to buy the necessary parts at local electronic-parts suppliers. For several of the projects (marked in column five), Vernier Software sells "parts kits."[3] The last column is an estimate of the approximate cost of the necessary components. This is the cost assuming that you need to buy all of the parts new.

The chart also identifies projects that have prerequisites—projects that cannot be done until another project is completed. For example, you cannot use the Computer-Controlled Car until you have built the Optically-Isolated Switches. In the chart, a circle after the project name indicates a prerequisite. The arrow points to the project requiring prior completion. Notice that two projects (Voltage Monitor and Optically-Isolated Switches) are each prerequisites for three other projects.

All of the circuits in this book have been used extensively at Vernier Software and at our test sites. It is unlikely that you would damage your computer in attempting to build or use any of the circuits in this manual. However, you should take care to double-check your wiring before you connect the circuit to the computer. Vernier Software shall have no liability or responsibility to any person or entity with respect to any liability, loss or damage caused or alleged to be caused during construction or use of any of these circuits.

---

[3] Details on the materials included in each parts kit are explained in the "Project Requirements" for the project. An Parts Kit Order Form is included at the back of this book.

PROJECT	DIFFICULTY FACTOR (1=EASY 10=HARD)	APPROX. ASSEMBLY TIME (HOURS)	PARTS AVAILABLE LOCALLY ?	VERNIER SOFTWARE PARTS KIT AVAILABLE?	APPROX. COST OF PARTS
1. Photogate Timer	3	2	YES	YES	$8
2. Reaction Time	2	2	YES	NO	$4
3. Microphone/Amplifier	5	4	NO	YES	$12
4. Humidity Meter	4	2	NO	YES	$23
5. Resistance/Capacitance Meter	3	2	YES	NO	$5
6. IC Temperature Probe	2	2	NO	YES	$13
7. Voltage Monitor	5	4	NO	YES	$24
8. Thermocouple	6	3	NO	YES	$13
9. pH Meter	6	3	YES	NO	$10 *
10. Strain Gage	10	6	NO	YES	$30
11. Switches	9	6	YES	YES	$30
12. Car	4	3	NO	NO	$20
13. Stepper Motor	2	1	NO	YES	$8
14. A Better Mousetrap	4	3	NO	NO	$5 †

* ALSO REQUIRES A pH ELECTRODE

† ALSO REQUIRES THE COMPONENTS FROM # 1 (PHOTOGATE TIMER) AND # 13 (STEPPER MOTOR)

NOTE TO TEACHERS

We hope this book can provide a start to a number of independent research or "science fair" projects. The core projects are explained in considerable detail and should be within the capabilities of most students. Each project has been assembled by several test assemblers, some of whom had very little experience in electronics. The "Project Requirements" section at the start of each project gives our estimate of the background experience needed for successful completion of the core project. Each project is left open-ended, with a number of different suggestions given for further work. Students who are interested in programming can improve the simple programs on the *Project Programs* disk. Students who are interested in electronics can often make changes or additions to the circuitry. A third type of project extension is to do further experimentation using the project as a tool. The student will have a chance to show some creativity, initiative and thoroughness with these project extensions.

Appendix G of this book is a short *Teacher's Guide*. It includes comments on the difficulty and complexity of the projects. It also includes results that we obtained in our tests of these projects.

THE *PROJECT PROGRAMS* DISK

Programs to use with each of the projects are provided on the *Project Programs* disk, which is attached to the inside back cover of this book. These sample programs are short and simple. In many cases they are missing features that any good program should have (such as rounding off of numbers or provisions for an easy way to quit). The programs are provided in this condition for two reasons: (1) We did not want to obscure the essentials of the program with a lot of nonessential material; and (2) The "missing features" can be added as part of your project extensions.

The *Project Programs* disk uses 5.25 inch, DOS 3.3 format. It is not copy protected. It is copy-righted, but you are permitted to make copies for your own use. We encourage you to make a backup copy of this disk immediately.

Appendix D is a list of the programs and files used with each project. If you want to convert the diskette to ProDOS, refer to *Appendix F* for instructions. The programs were written so that the conversion can be done easily.

The *Project Programs* disk also includes a program named UPDATE. This program will display all of the known corrections that should be made to this book. Run the UPDATE program as soon as you receive this book and correct any errors that are listed. If you find errors in the book that are not mentioned in the UPDATE program, please contact us. A *Feedback Form* is included in the back of this book.

If you are a novice at using a computer, you may find the following helpful:

- The <CAPS LOCK> key (on a IIe only) should always be depressed when you use the programs on the *Project Programs* disk.
- To start up the *Project Programs* disk: Place the disk in the disk drive labeled #1 and turn on the computer.
- To see a list of all of the programs on the disk: Type CATALOG and then press the <RETURN> key.
- To start a particular program: Type RUN (program name) and then press <RETURN>.
- To stop a program: Press the <CONTROL> and <C> keys at the same time and then press <RETURN>.
- To look at all of the lines of a program: Stop the program, type LIST and then press <RETURN>.
- To look at a particular line of a program: Stop the program, type LIST followed by the line number and then press <RETURN>.
- To change a particular line of a program: Stop the program, type the line number followed by the entire (changed) line and then press <RETURN>.

ELECTRONIC COMPONENTS

This chapter contains a quick look at the "building blocks" of the electronic circuits described in this book. It provides background information that should prove helpful when selecting and using components. It also reviews the symbols used in schematic diagrams. Obviously, this chapter cannot cover all aspects of electronics. The information is designed to help get you ready to do the projects in this book. For more information on electronics, you may wish to refer to some of the recommended electronics texts listed in *Appendix C*.

Before we begin this discussion, we should explain the use of two terms that sometimes cause confusion. The term *current* is used in this book to mean the flow of **positive** charge. Science texts sometimes use a different definition of the term current. The term *voltage* is used instead of the more technically correct term *electrical potential difference*. Unless otherwise specified, voltage is measured relative to ground. With both "current" and "voltage," we are using the standard terminology used in electronics.

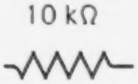
RESISTORS

Resistors are devices that limit the current in a circuit. There are two major categories of resistors: *fixed-value resistors* and *variable resistors*.

Fixed-Value Resistors

Every project in this book uses fixed resistors. These resistors have a specific resistance value in units of ohms. The symbol for ohms is the Greek letter capital omega: Ω. The term kilohm (kΩ) is used for a thousand ohms, and Megohm (MΩ) is used for a million ohms. The value of a fixed resistor is most often labeled on the resistor by a color code. This color code is explained in *Appendix B*.

On schematic diagrams, the symbol for a fixed resistor is:



The resistance value (10 kΩ in the diagram above) is indicated next to the resistor symbol. Another specification often quoted for resistors is their *power rating*. This is a measure of how much power they can handle without overheating. In general, the physical size of the resistor increases as the power rating increases. One-fourth (1/4) watt-resistors are large enough for all of the projects in this book.

There are actually two different types of fixed resistors that we will use in projects in this book:

Carbon-Composition Resistors

These are the standard, inexpensive resistors that are seen everywhere. They are usually labeled with four colored bands that identify their resistance value and their tolerance (the amount by which they may differ from their labeled value). See *Appendix B* for the resistor color code. In most cases, these resistors are guaranteed to be within 5% of their labeled value.

Metal-Film Resistors

These resistors are slightly more expensive, more accurate, and more stable. They are usually labeled with five color bands to indicate their resistance value and their tolerance. In most cases, metal-film resistors are guaranteed to be within 1% of their labeled value. These resistors also maintain their resistance value very well when their temperature changes. This type of resistor will be recommended when the resistance value must be controlled very accurately.

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☒ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

- Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

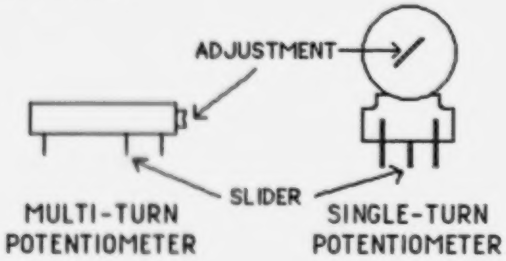
"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY

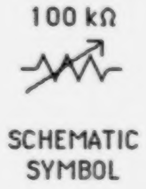
David L.
Verzius

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

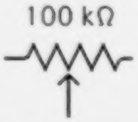
Variable Resistors

Many projects require a resistance that can be easily changed. Such a resistor is referred to as a *potentiometer*. A potentiometer has three terminals. Two of the terminals are at the opposite ends of the device. Between the two end terminals is a *slider*, which can move to different positions. In most applications, one of the end terminals and the slider terminal are used. The resistance between these two terminals varies as the position of the slider changes. Two types of potentiometers are illustrated below. The schematic diagram symbol is also shown. The value labeled is the maximum resistance of the potentiometer.





There are a few situations in which all three terminals of the potentiometer are used. In these cases, a slightly different schematic symbol is used, as shown below. The arrow indicates the slider terminal.



Other specifications that can be given to a potentiometer are:

Power rating: This is the number of watts of power the potentiometer can safely handle without overheating.

Taper: Some potentiometers change resistance in such a way that the resistance between the slider and one end is proportional to the distance from that end. This type of potentiometer is labeled *linear taper*. Other potentiometers change resistance in a more complicated way. These are labeled *audio taper* potentiometers. Either type is satisfactory for the projects in this book.

Mounting: Potentiometers that are designed to be soldered to a circuit board are called *PC mount*. The potentiometers illustrated above are PC mount. Potentiometers that are designed to be mounted in a hole drilled in a flat surface are called *panel mount*.

CAPACITORS

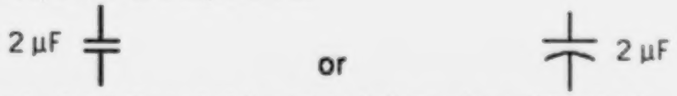
Capacitors are devices that store electrical charge. They are available in many different types, including electrolytic, tantalum electrolytic, ceramic, polyester, mica and polycarbonate. Each of these capacitor types differs in materials used and in construction. They also differ in cost, capacitance value, stability and physical size. One important difference is that some capacitors are *polarized*. Polarized capacitors have labeled positive and negative leads. When placing polarized capacitors in circuits, be sure to orient them in the correct way. Other capacitors are nonpolarized and may be oriented either way.

The two most important specifications about a capacitor are its *capacitance* and its *voltage rating*. Capacitance is a measure of how much charge a capacitor can store per volt. It is measured in a unit called a *farad*.[4] Most capacitors have values that are much smaller than a farad, so the most commonly used units for capacitors are microfarads (µF), which are equal to 10^{-6} farads, or picofarads (pf), which are equal to 10^{-12} farads. The other important specification for a capacitor is its voltage rating, which is the maximum voltage that the capacitor can handle without breaking down. When selecting a capacitor, make sure that it has at least the voltage rating required. It does not hurt to use a capacitor with a higher voltage rating.

---

[4] A capacitor has a capacitance of 1 farad if it stores 1 coulomb of charge when the voltage across it is 1 volt.

The schematic diagram symbol for a capacitor is:

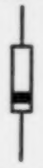


If the capacitor is polarized, a "+" symbol is used to indicate the positive lead.

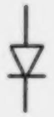
DIODES

Diodes act as "one-way valves" for electrical current. The two most important specifications used with diodes are the *current rating* and the *peak inverse voltage rating*. The current rating is the largest current that can safely go through the diode (in the direction the diode allows current to go). The peak inverse voltage rating (PIV) is the largest voltage that the diode can have across it without breaking down and conducting in the reverse direction.

Diodes are typically marked with a band on one end marking the negative end. The schematic diagram symbol for a diode looks like an arrow pointing in the direction that **positive** current flows. The diagram below shows a typical diode and its corresponding schematic symbol.



ACTUAL DIODE

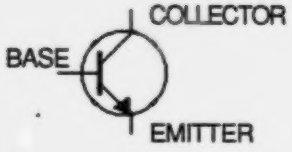


SCHEMATIC SYMBOL

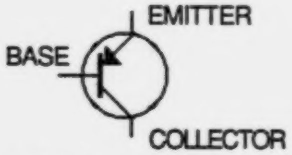
TRANSISTORS

There are hundreds of different types of transistors. They are used as amplifiers, in oscillators, in power supplies and in almost any complicated electronic circuit. For the projects in this book, the main use of transistors is as an electronic switch. The transistor conducts electricity between its *collector* and *emitter* terminals whenever current enters (or leaves) its *base* terminal. The types of transistors used in projects in this book are:

Bipolar NPN Transistor



Bipolar PNP Transistor



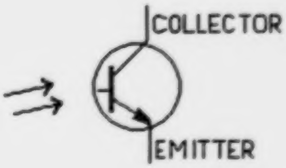
A little trick to use to keep these two transistor symbols straight is:
NPN stands for **N**ot **P**ointing i**N**, and
PNP stands for **P**ointing i**N**,

both referring to the direction of the arrow on the diagram. Notice that the arrow always indicates the emitter of the transistor.

Identifying the leads on a transistor can be a little tricky. We will include diagrams in the "Step-by-Step Instructions" showing how to identify the base, emitter and collector leads on the transistors required.

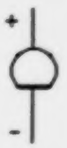
Phototransistor

Phototransistors are similar to the transistors discussed above, but their base terminal is not used. Instead, light striking the transistor acts to turn on the transistor. The schematic diagram is shown below. Refer to the manufacturer's data sheets to identify the emitter and collector leads on a phototransistor.

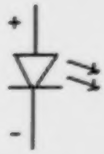


LEDs

LED stands for *Light-Emitting Diode*. They are just that—diodes that give off light. The light emitted can be red, green, yellow or infrared. LEDs should always be oriented properly. The negative lead of the LED is typically marked by a flat spot on the plastic case. The symbol for an LED is the same as the symbol for a normal diode, with small arrows indicating the light emitted.



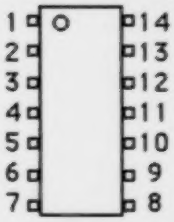
ACTUAL LED



SCHEMATIC SYMBOL

INTEGRATED CIRCUITS

Integrated circuits (ICs) are complex, miniaturized circuits designed for a particular task and sold as a single unit. ICs can contain dozens (or even hundreds) of resistors, capacitors, diodes and transistors. Most of the integrated circuits used in these projects are packaged in *DIP* packages. DIP stands for "Dual In-line Pins," indicating that the connection pins are in two parallel rows. For all ICs of this type, the pins are numbered in a counterclockwise direction as seen from the top (with the pins pointing away from you). The #1 pin is marked with a dot nearby or sometimes with a large notch in the case. A top view of a typical 14-pin DIP IC, with the pins numbered, is shown below:

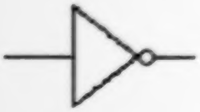


TYPICAL DIP IC (TOP VIEW)

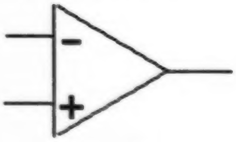
When working with integrated circuits, be very careful about identifying the pins. Remember that the illustration above is a top view. If you are soldering on the underside of the IC, the pins appear numbered in a clockwise fashion.

When DIP integrated circuits are needed for these projects, IC sockets will always be recommended. The socket is soldered into the circuit, and the IC is plugged into the socket after all of the soldering is completed. A socket provides two advantages: (1) It allows you to easily change ICs if there is ever a need to do so; and (2) The IC is kept away from the heat of the soldering iron. When you place the IC in the socket, be very careful to place pin #1 of the IC into the proper hole. Also, avoid bending and breaking the thin pins of the IC.

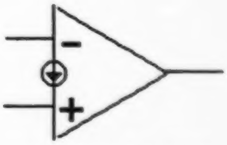
The schematic diagram of an IC depends on the internal circuitry contained in the IC. ICs are used as building blocks in modern electronics, and often it is not necessary to understand how the IC operates internally as long as you know what the IC does. Three specific types of integrated circuits that are used in the projects are shown in the diagrams below:



INVERTER



OP AMP



NORTON OP AMP

In the "Step-By Step-Instructions," we will include *working diagrams* of how to connect the IC. These diagrams show the IC package with the pins numbered and the appropriate connections indicated.

WIRE

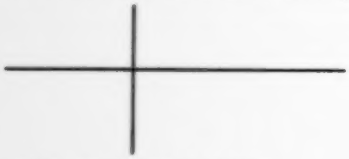
Several terms are used to describe wire used in these projects:

x-conductor wire: The "x" refers to the number of separate wires that are packaged together.

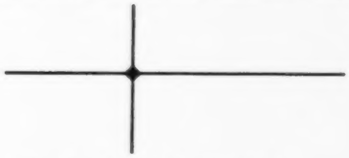
shielded cable: This consists of wire(s) that are contained inside a "tube" usually made of braided, copper wire.

stranded wire: Stranded wire is made of many fine, metal wires twisted or braided together. Stranded wire is much more flexible than a solid wire of the same size and is recommended for the projects in this book.

The diagram below show the schematic symbols for wires that are crossing with and without electrical connection between them. These symbols can sometimes cause confusion if not understood.



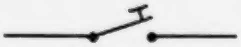
TWO WIRES CROSSING
NOT CONNECTED



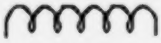
TWO WIRES CROSSING
CONNECTED

OTHER COMPONENTS

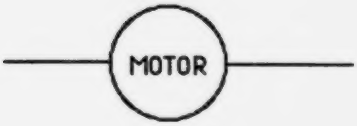
Other symbols used on schematic diagrams in this book are shown below:

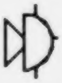


SWITCH

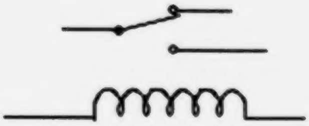


COIL OF WIRE
OR ELECTROMAGNET

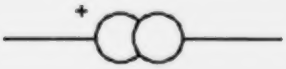




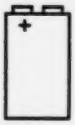
MICROPHONE



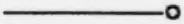
RELAY



CURRENT SOURCE



9-VOLT BATTERY



CONNECTION POINT

SOURCES OF ELECTRONIC COMPONENTS

A parts list is provided at the beginning of each project. This section gives suggestions on where to purchase electronic components.

Note: Electronic-parts suppliers frequently change part numbers and discontinue parts. At the time of publication, all references to part numbers were correct. We are sure this will not be true for long. The program UPDATE on the *Project Programs* disk will display any known changes in part numbers or availability at the time we ship the book to you.

MAIL-ORDER PARTS SUPPLIERS

Before assembling any of the projects in this book, we recommend writing for free catalogs from a number of mail-order, electronic-parts suppliers. These companies have a wide variety of parts and usually offer low prices. Even if you do not purchase your components via mail, their catalogs will provide a good source of information about what is available, and they will give you an idea of what you should expect to pay if you purchase the components locally. A list of a few parts houses is provided below:

Mouser Electronics
2401 Hwy 287 North
Mansfield, TX 76063
(817) 483-4422
or
Mouser Electronics
11433 Woodside Ave.
Santee, CA 92071
(619) 449-2222

All Electronics
905 S. Vermont Ave.
P.O. Box 20406
Los Angeles, CA 90006
(213) 380-8000

Digi-Key Corporation
P.O. Box 677
Thief River Falls, MN 56701
(800) 344-4539

Jameco Electronics
1355 Shoreway Road
Belmont, CA 94002
(415) 592-8097

BNF Enterprises
119 Foster St.
P.O. Box 3357
Peabody, MA 01961-3357
(617) 531-5774

RADIO SHACK STORES

Radio Shack stores are located in almost every community. They provide a convenient source for many components. In some cases Radio Shack sells components that are not available from other sources. We list Radio Shack part numbers for some of the parts used for these projects. Unfortunately, Radio Shack **often** changes part numbers or discontinues a product.

LOCAL ELECTRONIC-PARTS SUPPLIERS

If you live in a large town, check your telephone book for "Electronic Equipment and Supplies." You may find a local dealer who can supply components at very good prices. On the other hand, be aware that some of these businesses will not want to deal with small purchases. They may have large minimum orders, or they may not have very many components in stock. Hamilton-Avnet is one large parts distributor with offices in many locations. They are sometimes willing to sell small quantities of components.

VERNIER SOFTWARE PARTS KITS

Because several of the projects in this book require components that are not readily available at local electronic-supply houses, we sell parts kits for some projects. Each parts kit contains all the electrical components for the project. On projects that connect directly to the computer, a 16-conductor ribbon cable is included. The parts kits do not include tools, supplies or solder. The following parts kits are now available:

Project 1	Photogate Timer	$8.00
Project 3	Microphone/Amplifier	$12.00
Project 4	Humidity Meter	$23.00
Project 6	IC Temperature Probe	$13.00
Project 7	Voltage Monitor	$24.00
Project 8	Thermocouple	$13.00
Project 10	Strain Gage	$30.00
Project 11	Optically-Isolated Switches	$30.00
Project 13	Stepper Motor	$8.00

Prices and availability are subject to change. These parts kits can be ordered from:

Vernier Software
2920 S.W. 89th. Ave.
Portland, OR 97225
(503) 297-5317

Include $3.00 per order for shipping. An order form can be found at the back of this book.

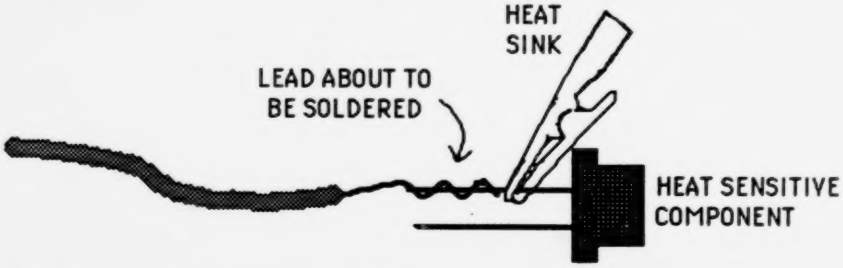
PROJECT ASSEMBLY TIPS

SOLDERING VERSUS PROTOTYPING BOARDS

Each of the projects in this book involves assembling an electronic circuit. In each case the "Step-by-Step Instructions" suggest soldering the circuit together. We recommend soldering because it produces a finished product that you can use for some time to come. If you would like to try out the circuit before soldering it together, you can assemble the circuit on a solderless "breadboard" (for example, Radio Shack #276-175, #276-174 or #276-169). Using a breadboard is a good idea if you plan to make major changes in the circuit at a later time or if you do not plan to use the assembled circuit for very long. The disadvantages of prototyping boards are that they are expensive, and the poor connections you sometimes get can cause frustrations. There is one project (Project 4, *Humidity Meter*) that will not work properly if assembled on a prototyping board.

SOLDERING TIPS

- Use a good quality, "pencil-type" soldering iron. Soldering irons with temperature-regulated tips are the best (but expensive). If a temperature-regulated iron is not available, use a 30-watt or lower power iron with a fairly narrow tip. **Do not use a large, high-wattage soldering iron or gun for these projects.**
- Keep the tip of the iron clean by periodically wiping it on a damp sponge. The tip should always look bright and shiny.
- **Use narrow-gauge, rosin-core solder.** Thin solder melts quicker and is easier to control. Solder with a diameter larger than .050 inches is difficult to use when assembling circuit boards. Never use acid-core solder. The acid will cause corrosion of metal wire and other components.
- Make sure the components are clean and free of corrosion before soldering. Lightly rubbing a corroded lead with steel wool usually cleans it.
- When soldering any component that can be damaged by overheating (in particular, transistors, LEDs and integrated circuits[5]), use a *heat sink*. A pair of pliers, an alligator clip or any metal object can serve as a heat sink. The idea is to provide a large metal object into which heat can flow so that the heating of the component is reduced. Place the heat sink between the solder and the component, as shown in the figure below:



- Make a good physical contact between the parts to be assembled (by twisting wires, etc.) before you start to solder.
- Always remember the old saying, "**Heat the metal not the solder.**" Place the heated tip of the iron against the parts to be soldered. Try to maximize the contact area between the tip and the metal. Be patient. After a few seconds, move the solder in and let the metal heat up the solder. Wait until the solder flows and looks like mercury.
- Move the iron away and hold everything steady for a few seconds. Inspect the connection. It should be bright and shiny. If it looks rough and dark, it is not a good connection. There may not even be electrical connection between the parts. Reheat the joint and try again. Check the

---

[5] When using integrated circuits you can avoid overheating problem by using IC sockets. Solder the socket in place and then later plug in the integrated circuit.

connection by trying to pull the objects apart. If you feel any looseness, it may not be a good electrical connection.

- Check for *solder bridges*. Sometimes when you solder on a circuit board, solder can connect across places that should not be in electrical contact. If necessary, reheat the connection and remove the extra solder.
- Clip off excess lead wires near the solder junction. If these leads are not cut short enough, they can bend over and create a short.

OTHER CONSTRUCTION SUGGESTIONS

- Avoid using solid wire; instead use stranded wire (made up of many thin wires, twisted together). Solid wire will break if it is bent back and forth very many times. Stranded wire is more flexible and can withstand bending much better. When using stranded wire, however, watch out for stray strands that can cause a short.
- Minimize wire movements. When wire leads are allowed to move back and forth, there are usually problems. This is especially true when a wire is allowed to bend at the point where it is soldered. Fasten wires down by tying them to something solid or by using clamps or glue.
- Be careful about static electricity when handling ICs, especially CMOS ICs. Touch a grounded metal object before touching the chip.
- When working with a circuit board, it is sometimes helpful to hold the board up to a light so that you can look at the nonplated side and see the plating pattern on the other side.
- Put your assembled circuit in a plastic or metal box to protect it.
- Make your wiring neat. Keep leads as short as possible. Unnecessary loops of wire can cause problems by acting as antennae for electrical and magnetic fields. Neatness counts, even in electronics.
- Keep exposed, bare wire at a minimum. Even if the uninsulated parts of a wire are not touching anything, they may cause problems later. If the circuit gets dropped or jolted, the wire may accidentally touch something it should not touch. Keep leads short and use plastic tape or heat-shrink tubing as an insulator.
- When you think the project is complete and correct, take a few minutes to double-check the circuit before applying the power.
- For the first test of a circuit, briefly turn on the power and look, smell and feel (carefully) for any overheating. If any component is hot, turn off the power. There must be something wrong.

USEFUL TOOLS

- Heat-shrink tubing is very useful for insulating the solder joint between two wires. Heat-shrink tubing is plastic tubing that shrinks to about half its original diameter when heated. Slip the tubing over one of the wires to be soldered. After the soldering is completed, move it over the solder joint. Heat the tubing with a match or a hot-air gun, and it neatly shrinks down over the solder joint. Heat-shrinking tape (Radio Shack #64-2335) is also available for insulating connections.
- If you need to remove a soldered component from a circuit board, you will probably find *solder wick* helpful. Solder wick is braided copper wire that is placed over the solder to be removed and then heated. The solder is drawn into the copper as it melts. There are also devices that use a vacuum to suck up molten solder.
- "Helping Hands" (Radio Shack #64-2093) are helpful for holding components while soldering.

TROUBLESHOOTING

Troubleshooting is the process of trying to find the problem when the circuit does not work right. Anyone who builds electronics projects will need to do some troubleshooting. Some suggestions on how to approach troubleshooting are given below:

◊ Meters to use in troubleshooting

A *multimeter* is an invaluable tool for troubleshooting electronic circuits. A multimeter is a single meter that can be used as an ohmmeter (to measure resistance), a voltmeter (to measure electrical potential differences) and an ammeter (to measure currents). All three of these functions can be helpful in troubleshooting. We recommend that you obtain a digital multimeter, if possible. These meters have extremely ***high impedance***, which means they do not draw much current from the circuit they are testing. This allows them to be used in places where a conventional meter will not work. There are several troubleshooting tests mentioned in this book that are only possible with a high-impedance meter.

◊ Troubleshooting techniques

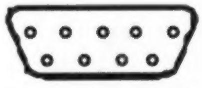
Use a logical, organized approach to your troubleshooting. Try to "zero in" on the part of the circuit that is not working. Specific troubleshooting suggestions are given for each project in this book. The suggestions below are general, and apply to all of the projects:

1. First, check for some of the "easy-to-make" mistakes, such as:
   - If you are using an integrated circuit, is the chip inserted the wrong way or are some of the lead pins missing the socket?
   - Are the leads from the Apple game port used properly? If you are using a 16-pin connector to the internal game port, are you sure that you have the plug inserted with pin #1 toward the keyboard?
   - Is everything that is supposed to be soldered really soldered?
   - If you are using a circuit board, are there any solder bridges connecting areas that should not be connected?
   - If you are using any components that have identified leads (such as polarized capacitors, transistors, LEDs or diodes), is the component oriented in the correct way?
2. If the circuit still does not work, force yourself to go over it again, wire by wire. If you have worked with the circuit for a long time, you may be too familiar with it. It is similar to proofreading your own writing after you have worked on it all day. Take a break and come back to it later or get a friend to look over the circuit.
3. Resist the urge (it's natural) to assume that one of the components that you bought is faulty or that the instructions are wrong. These things are sometimes to blame, but wiring errors are much more likely.
4. Try reheating every solder joint. Sometimes a connection that appears to be soldered really is not. By reheating the connection, you may fix a "cold-solder joint" that you did not even know about.
5. Double-check your wiring by using an ohmmeter. Make sure the points that are supposed to be wired together really are. The resistance between them should be nearly zero. These tests should be done with the circuit disconnected from the computer. They can help you to locate incorrect wiring and to identify solder joints that appear to be connected, but are not.
6. Try to "zero in" on the part of the circuit that is not working by using a voltmeter to check the voltages at various points on the circuit. This test should be done with the circuit connected to the computer and the power on. Check, for example, to see if all of the points that are supposed to be connected to the Apple's +5V line are really at 5 volts? Try to use the meter to identify which parts of the circuit are working and which are not. Then, think about what possible mistake in the circuit could cause the symptoms you have noticed. Triple-check the things that seem most suspicious.
6. Refer to the specific troubleshooting suggestions given for each project.

THE APPLE GAME PORT

One of the things that has made the Apple II+ and IIe computers so popular is the game port. The game port provides several different ways of getting information into and out of the computer. Actually there are two different game ports that are used on Apple II+ and IIe computers. Apple II+ computers have an internal 16-pin DIP game port. The Apple IIe has this same internal game port, but in addition has a second, external game port.[6] This second game port has only nine connections on a sub-D connector.[7] Both game ports are shown below:

APPLE GAME PORT
SOCKET
(INSIDE II+ OR IIe)

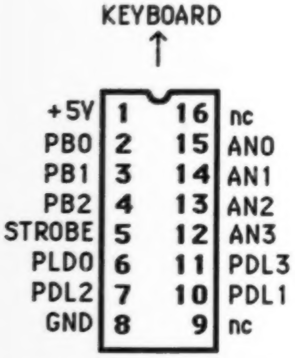


GAME PORT SOCKET
(OUTSIDE BACK OF IIe)

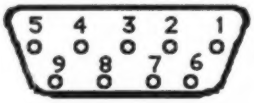
The input and output connections of the game ports fall into six different categories:

- power-supply lead (+5V)
- ground lead (GND)
- switch inputs (PB0, PB1, PB2)
- paddle inputs (PDL0, PDL1, PDL2, PDL3)
- annunciator outputs (AN0, AN1, AN2, AN3)
- strobe output (STROBE)

Only the first four types of connections are available on the external 9-pin game port on the IIe. The diagrams below show the pin connections for both game ports. *Appendix A* discusses cables that can be used to connect to the two Apple game ports. Each of these game-port connections will be discussed below.



INTERNAL GAME PORT



EXTERNAL GAME PORT

1 = PB1
2 = +5V
3 = GND
4 = PDL2
5 = PDL0
6 = PB2
7 = PB0
8 = PDL1
9 = PDL3

---

[6] The Apple IIc computer has a 9-pin game port similar to this; however, the PDL2, PDL3 and PB2 lines are not available on the 9-pin Apple IIc game port. The PB0 and PB1 input lines on the IIc are a little different electrically from the switch input lines on the IIe. Circuits that work with the IIe switch inputs will not always work using a IIc.

[7] The name comes from the "D" shape of the connector. The original D connector had 25 pins; those with fewer pins are called "sub-D."

POWER-SUPPLY LEAD (+5V)

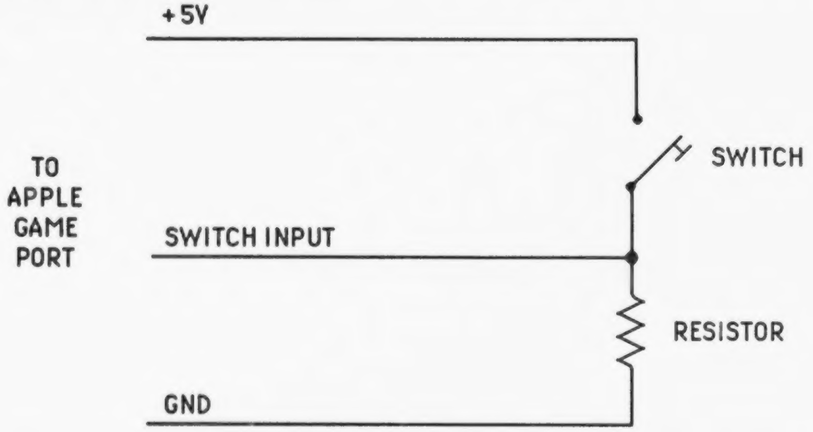
Either of the game ports has a lead labeled +5V. This lead is connected to the +5 volt output of the Apple power supply. It can provide reference voltage levels for use in switches, or it can supply a small current for use in powering electronic devices. **The current drawn from the +5V line should never exceed 100 milliamperes.**

GROUND LEAD (GND)

Both the internal and external game ports have a lead labeled GND, which is electrically connected to the ground of the computer's power supply.

SWITCH INPUTS (PB0, PB1, PB2)

There are three switch input connections available on either the internal or external game ports of an Apple II+ or IIe. The lines are labeled PB0, PB1 and PB2. They are *digital input lines*. This means that the computer can sense which of two possible conditions (low voltage or high voltage) is present on the line. The switch input lines are usually connected to the push-button switches on a game controller or joystick in a circuit similar to this:



In the diagram above, the voltage on the switch input line would be low if the switch were open, because the line is connected to ground through the resistor. If the switch is closed, the switch input line is connected to the +5 volt line, and the input voltage rises.

The computer can sense whether the voltage at the switch input line is low or high. A particular memory location is changed automatically as the voltage level at the switch input changes. The memory bytes used are:

Switch Input	Memory Location[8]
PB0	49249 ($C061)
PB1	49250 ($C062)
PB2	49251 ($C063)

The number in each of these memory locations depends on the voltage at each of the three switch input lines. If the voltage at the switch input line is higher than 2 volts, the corresponding memory location will contain a number larger than 127. If the voltage on a switch input line is less than 0.8 volts, then the value stored in the corresponding memory location will be less than 128. If the voltage on the switch input line is in the range 0.8 to 2.0 volts, it may be interpreted either way.[9] One interesting thing about the way the switch inputs work is that the exact number stored in the switch input memory location is random. The voltage on the switch input line controls only whether the number is greater than 127 or not.

---

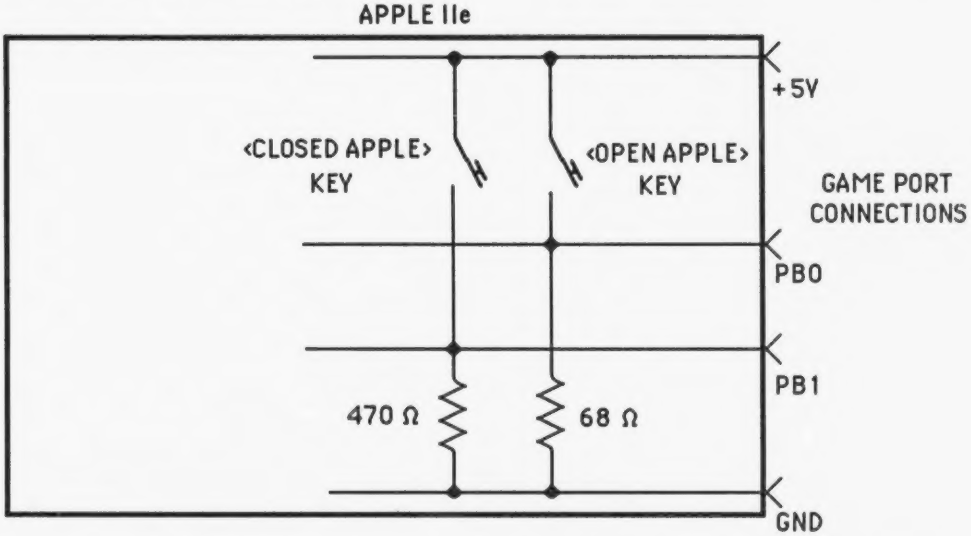
[8] The numbers in the parentheses are the memory addresses in hexadecimal or base-16 notation. This number system is often used with computers. Hexadecimal numbers in this book are written preceded by a dollar sign to distinguish them from decimal numbers.

A PEEK command in Applesoft BASIC can be used to check the status of a switch input line. Write a PEEK statement to determine the number stored in the memory location corresponding to the switch input line. Check whether or not this value is greater than 127. The statements below are examples:[10]

```
IF PEEK(49249) > 127 THEN PRINT "PB0 LINE IS HIGH"
IF PEEK(49250) < 128 THEN PRINT "PB1 LINE IS LOW"
IF PEEK(49251) <= 127 THEN PRINT "PB2 IS LOW"
```

The BASIC program PB.TESTER on the *Project Programs* disk displays the status of the three switch input lines.

Note for Apple IIe Users Only: The PB0 and PB1 switch input lines on an Apple IIe are also connected to the <OPEN APPLE> and <CLOSED APPLE> keys. This causes these two input lines to respond differently on the Apple IIe as compared to the Apple II+. The two "APPLE" keys on an Apple IIe are connected as shown below:



APPLE IIe <OPEN APPLE> AND <CLOSED APPLE> KEYS CIRCUIT

The resistors between the switch input lines and GND limit the way the PB0 and PB1 switch input lines can be used. Because of these resistors, some circuits that work using Apple II+ switch inputs will not work using IIe switch input lines. The PB2 line on an Apple IIe is no different from an Apple II+. For this reason, most of the projects in this book that use switch inputs use the PB2 line. This avoids complications caused by the extra resistors.

---

[9] In electronics terminology, switch input lines are referred to as low-power Shottky (LS) TTL inputs.

[10] The status of the switch input lines can also be checked using machine language. See *Appendix E* for examples.

Using the PB0 and PB1 lines on the IIe can also cause unexpected results in some cases. The most frequent problem involves the Apple's "automatic self-test." If the PB1 line is connected to a voltage higher than 2 volts, the IIe responds as if the <CLOSED APPLE> key were held down. The Apple IIe has a built in "self-test" routine that is started automatically each time the <CLOSED APPLE> key is held down when the computer is first turned on (or rebooted). During the self-test, the video monitor goes dark and the computer goes through several seconds of diagnostic tests. When the test ends, a status message is reported.

Turning on (or rebooting) the Apple IIe computer while a voltage signal is being fed into the PB1 lead can accidentally initiate this self-test. This can be very frustrating to the user. If the <OPEN APPLE> key is held down when the self-test is started, a high-pitch sound is emitted by the Apple's speaker during the test. A voltage signal connected to the PB0 lead can cause this effect. We avoid the "self-test" problems in these projects by not using the PB1 input line.

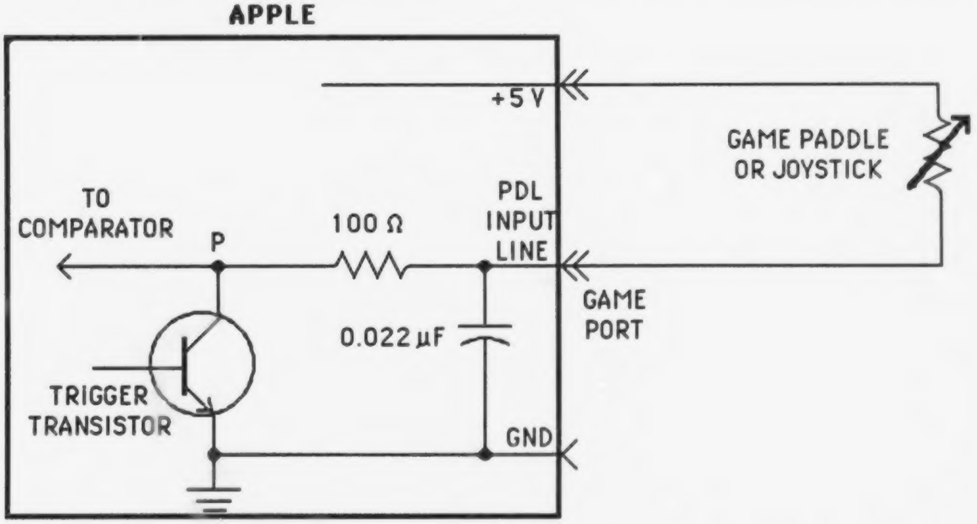
PADDLE INPUTS (PDL0, PDL1, PDL2, PDL3)

The PDL input lines were originally designed to be used to read the position of a game paddle (hence the label PDL) or joystick. There are four of these PDL lines on the Apple II+ and IIe game ports. They are labeled PDL0, PDL1, PDL2 and PDL3.

The game paddle or joystick that is used with the PDL inputs is really a variable resistor. Its resistance varies from 0 to 150 kilohms. It is connected between the +5V line and one of the PDL lines on the game port. Using a command built into Applesoft BASIC, the resistance can be determined by the computer. The four BASIC commands PDL(0), PDL(1), PDL(2), and PDL(3) produce an integer between 0 and 255 that depends on the resistance of the variable resistor. The number in the parentheses determines the PDL line being checked. The larger the number displayed on the monitor, the higher the resistance of the variable resistor. Typical BASIC statements using these commands are shown below.

```
100 PRINT PDL(0) : REM DISPLAY THE STATUS OF PDL0
200 X = PDL(1) : REM SET X ACCORDING TO STATUS OF PDL1
```

To understand what is really going on when you use a BASIC PDL(0) command, we need to take a more detailed look at the PDL input lines. A typical circuit using the PDL lines is shown below:



TYPICAL GAME-PADDLE CIRCUIT

The part of the circuit drawn inside the rectangle is inside the computer.[11] An indirect method is used to determine the resistance connected to the PDL line. To understand this method, you need to understand how the various components of the circuit function:

- **variable resistor**: The variable resistor typically varies in resistance from 0 to 150 kilohms. It limits the flow of current from the +5V terminal toward the capacitor according to Ohm's law. If the resistance is increased, the current to the capacitor is reduced. If the resistance is decreased, the current increases.
- **0.022-µF capacitor**: A capacitor is a device that stores electrical charge. As the charge stored on the capacitor increases, the voltage across the two leads of the capacitor (and at point "P" in the diagram above) increases.
- **trigger transistor**: The trigger transistor in this circuit acts as a switch that is controlled by the computer. When the trigger transistor is "on," current can pass through it. When it is "off," no electrical charges can move through it.
- **comparator**: The comparator is an electronic circuit that can sense the voltage level at point "P" in the circuit diagram above. The computer uses the comparator to determine if the voltage at point "P" is less than or greater than a specific voltage level (3.3 volts).

Under normal circumstances, the trigger transistor is in its "on" condition. This allows any charges that might be present on the positive side of the capacitor to flow through the 100-Ω resistor to ground. As a result, the capacitor is normally totally discharged. The voltage at point "P" is 0 volts. When the BASIC PDL( ) command is executed, the trigger transistor is switched "off." At the same instant, the computer begins keeping track of time and checking the voltage at point "P". The current from the +5V line through the variable resistor now acts to charge the capacitor. The voltage across the capacitor (and at point "P") begins to increase. The computer measures the amount of time it takes for the voltage at point "P" to reach a specific level (as determined by the comparator). A machine-language routine in the Apple's ROM[12] is used to do this. The time is really just a "count" of how many loops the program goes through before the capacitor is charged. If the variable resistor is set for low resistance, then the current is large and the capacitor will charge up quickly, which means a low count is registered. If the variable resistor is set for a high resistance, the current into the capacitor is small and the capacitor will take a long time to become charged, which means a high count will be indicated. If the capacitor is not charged even when the count reaches 255, the routine ends and a count of 255 is registered. Each loop takes 11 µsec (11 x 10^{-6} sec). The count is always a whole number in the range 0 to 255. For example, a count of 193 means that it took 193 x 11 µsec or about 2 milliseconds for the capacitor to become charged. A count of zero indicates that the capacitor charged almost immediately, so the current must have been very large.

As you can see, the PDL circuit and accompanying BASIC commands are an elaborate way for the computer to sense the resistance of an external resistor. The BASIC program PDL.TESTER on the *Project Programs* disk displays the status of the four PDL input lines.

Using special methods (see Projects 5 and 6, for examples), the PDL input lines can also be used as a way of measuring current or capacitance.

ANNUNCIATOR OUTPUTS (AN0, AN1, AN2, AN3)

The Apple II+ and IIe internal 16-pin game ports have four *annunciator output* connections. These four terminals are labeled AN0, AN1, AN2, and AN3. With simple BASIC statements, you can change the voltage at any of these four terminals from low (near 0 volts) to high (about 5 volts).[13] These terminals are **not** available on the Apple IIe 9-pin external game port.

---

[11] The circuit above is a PDL circuit for the Apple IIe. A PDL circuit on an Apple II+ has one small difference: The 100-Ω resistor is between the capacitor and the game port instead of between the capacitor and the trigger transistor.

[12] *ROM* is an abbreviation for "read only memory." It consists of memory chips in which computer code is permanently stored. This code can be read by the computer at any time, but it can never be rewritten.

The statements used to control the voltage at the annunciator terminals are shown in the table below:

Condition	BASIC POKE Statement
AN0 off (0 volts)	POKE 49240, 0
AN0 on (+5 volts)	POKE 49241, 0
AN1 off	POKE 49242, 0
AN1 on	POKE 49243, 0
AN2 off	POKE 49244, 0
AN2 on	POKE 49245, 0
AN3 off	POKE 49246, 0
AN3 on	POKE 49247, 0

Notice that the first number in each of these POKE statements is different for each line. The second number in the POKE statement is just a filler. A zero is used in all of the statements above, but any number could be used. It makes no difference, but there must be some number there.[14] The sample BASIC program below would turn the AN0 terminal on and off at regular intervals:

```
10 POKE 49241, 0 : REM AN0 ON
20 FOR J= 1 TO 2000 : NEXT J : REM STALL
30 POKE 49240,0 : REM AN0 OFF
40 FOR J= 1 TO 2000 : NEXT J : REM STALL
50 GOTO 10 : REM REPEAT
```

Be careful when using the annunciator lines. The Apple's annunciator lines cannot be used to power anything that requires very much current. If you try to draw too much current from an AN line or sink too much current into it, you could actually damage the computer.[15] The annunciator lines are designed to drive only the inputs of other integrated circuits.[16] If a larger load is to be controlled with these lines, a secondary "buffer" circuit must be used between the annunciator line and the load. Project 11 (*Optically-Isolated Switches*) is an example of how the annunciator output lines can be used to control useful devices.

STROBE OUTPUT

The 16-pin internal game port on an Apple II+ or IIe has an infrequently used strobe-output line. This connection is not available on the 9-pin external game port. The strobe-output line is normally at 5 volts. The voltage on the line drops briefly (for about 1 µsec) each time the memory location 49216 is used (either to read or to store a number). None of the projects in this book use this connection.

---

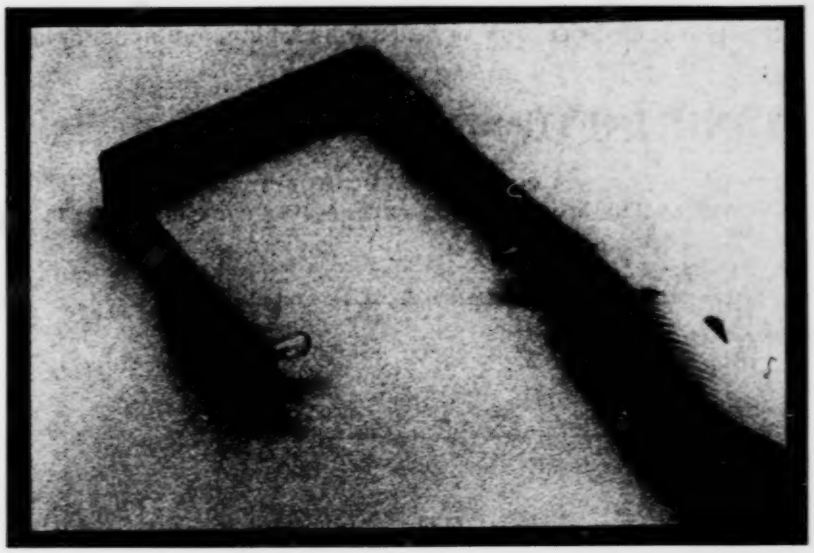
[13] The annunciator lines are digital outputs. This means that they can have one of two possible values. Sometimes computers are connected to analog outputs. Analog outputs can vary over a range of many possible voltages.

[14] This is a fairly common, but strange way of doing things on the Apple II. It is called a *soft switch.*

[15] As is the standard practice in electronics, by *current* we mean the movement of positive charge. When the annunciator output line is high (5V), it acts as a *source* of current. When the line is low (0V), it acts as a *sink* for current.

[16] The annunciator output lines on an Apple II+ are standard 74LS TTL outputs. The annunciator lines on an Apple IIe connect directly to the Input/Output Unit. The Apple IIe reference manual warns that these lines are to be used to drive only one TTL load.

PROJECT 1:
PHOTOGATE TIMER



In this project you will build and use a *photogate*. A photogate consists of a light source and a light detector. Whenever an object moves through and blocks the beam of light between the source and the detector, a signal is produced that can be detected by the computer. The computer can use this signal to start and stop timing operations. The computer serves as a very accurate clock. Times are measured to the nearest millisecond (0.001 second). With this system you can time a rolling object, a falling object, a pendulum, a camera shutter or a strobe light.

PROJECT REQUIREMENTS

Background Knowledge

- Electronics: Some previous experience with soldering is desirable.
- Programming: Some familiarity with BASIC is recommended for the core project. Some of the project extensions require experience with BASIC programming.

Parts

☐ NPN infrared phototransistor, OP598 or equivalent (Radio Shack usually sells a suitable phototransistor as RS #276-145. The identification of the leads of phototransistors can be confusing. Refer to the manufacturer's data sheets or packaging for information on how to identify the emitter and collector of the phototransistor.)
☐ infrared LED, OP297 or XC-880-A (RS #276-143) or equivalent
☐ support for photogate (metal or wood)
☐ cable to Apple game port (See *Appendix A*)
☐ 3.3-kΩ resistor, 1/4 watt, 10%
☐ 220-Ω resistor, 1/4 watt, 10%
☐ (2) pieces of wire 20 cm long

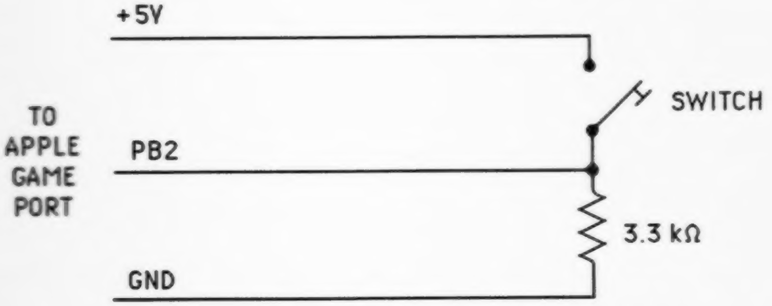
Note: A "Photogate Parts Kit" containing all of the components needed for this project, including a 16-conductor ribbon cable to the Apple game port, is available from Vernier Software. This kit sells for $8.00. An order form is provided at the back of this book.

Tools and Supplies

Solder, soldering iron, plastic electrical tape or heat-shrink tubing, multimeter, pliers, wire strippers and wire cutters.

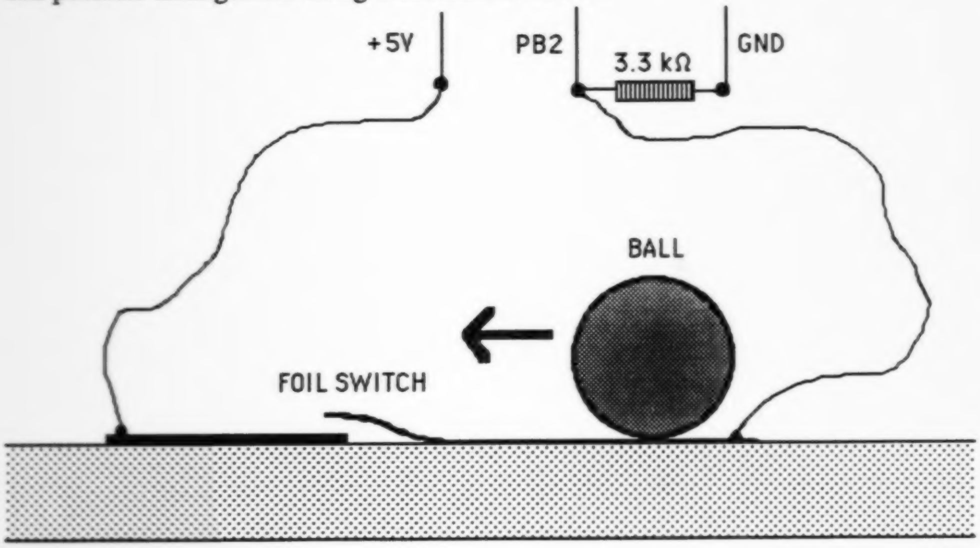
BACKGROUND INFORMATION

The Apple computer game port has three *switch input* terminals: PB0, PB1 and PB2. These terminals are usually connected to the push-button switches on a game controller in a circuit similar to this:



The computer can sense the voltage at the switch input line (PB2). In the diagram above, that voltage would be low if the switch was open, since the input line is connected to ground through the 3.3-kΩ resistor. If the switch is closed, the input line is connected to the +5 volt line and the input voltage rises.[1]

This idea can be used to build a very simple timing device for rolling objects. The diagram below shows one possible arrangement using a mechanical switch:

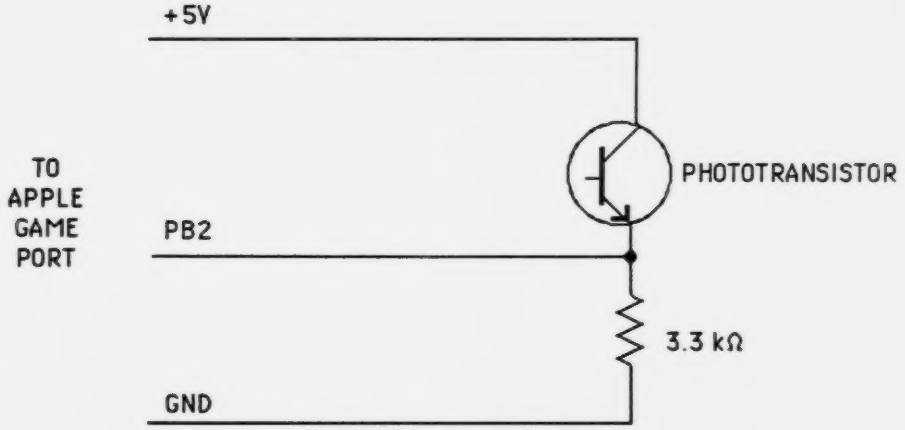


---

[1] Refer to "The Apple Game Port" section of the *Introduction* for more information about the switch input lines.

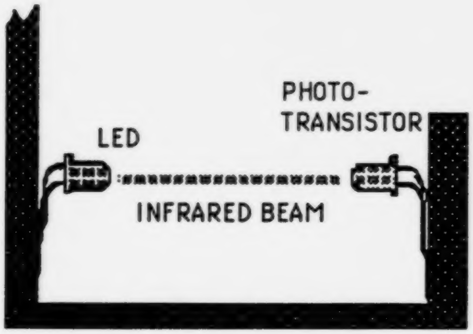
The ball rolls along on top of the foil and will bend down the foil to close the switch. This raises the input voltage on PB2. The computer can sense this change and a timing cycle can be started or stopped with this "timing gate." There are two possible problems with this kind of timing. First, the switch will slightly slow the motion of the object. Second, the foil switch tends to wear out or need frequent adjustment. A photogate eliminates both of these problems.

In a photogate, a phototransistor takes the place of the switch. The phototransistor circuit is shown below:

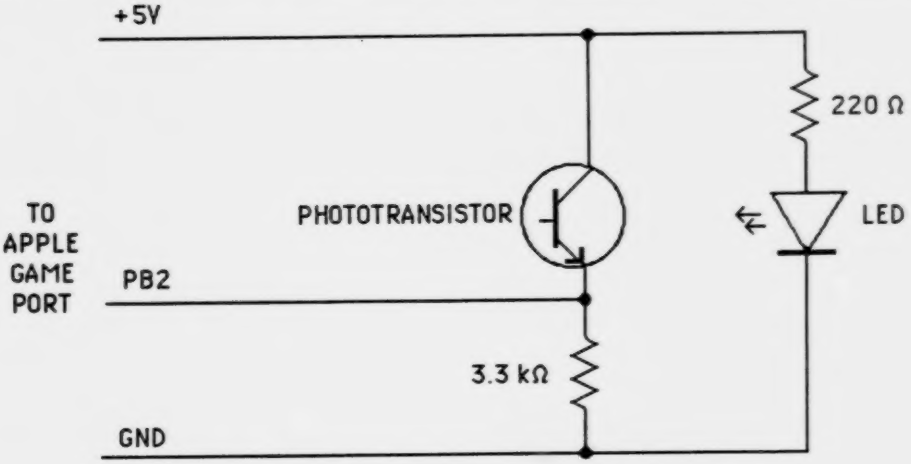


A phototransistor is basically a light–sensitive switch. It has very high resistance (over 100 kΩ) when it is in the dark. The resistance drops to only a few hundred ohms when light shines on it, and as a result, the voltage at the switch input increases just as it did when the mechanical switch was closed. This simple circuit thus allows the computer to detect when a light beam aimed at the phototransistor is interrupted by an object.

The light source shining on the phototransistor can be a normal light bulb, the sun, a laser, or almost any light source. It is best to have a narrow beam of light directed at the phototransistor. One convenient way to do this is to mount a small light source permanently near the phototransistor. Infrared light-emitting diodes (LEDs) are good choices for this purpose. They are inexpensive, durable and they provide a narrow beam of infrared light for the phototransistor to detect. Also, LEDs do not require much current, so we will be able to draw the necessary current from the Apple computer. Infrared light is "redder" (or of longer wavelength) than our eyes can see. Phototransistors, however, can detect this infrared light. A complete photogate with LED and phototransistor looks something like this:



The complete circuit for the photogate that you will build for this project is shown in the schematic diagram below. The 220-Ω resistor limits the flow of current to the LED to the proper level so that it does not overheat.



PHOTOGATE SCHEMATIC

With this system, the computer will be able to detect when an object moves through the space between the LED and the phototransistor. Whenever the beam is blocked, the phototransistor is "off" and the voltage signal into the computer is low. When the beam is unblocked, the voltage signal will be high, since the resistance of the phototransistor is much less than that of the 3.3-kΩ resistor.

The computer program to use with such a photogate needs to keep track of time while always checking the status of the photogate. As an example, consider measuring the time it takes an object to move completely through a photogate.[2] Before the object reaches the photogate, the phototransistor is conducting and the input signal is high. At first, the program needs to keep checking for a low–voltage signal on the switch input line, indicating that the beam is blocked. Then it should start keeping track of time. During this timing, the program should continuously check the switch input line to watch for a high–voltage signal, which would indicate that the object has moved out of the beam, and the timing should stop. When this is done, the program should display the time interval measured.

Here is a BASIC program that will do this job:[3]

```
100 CT = 0
200 IF PEEK(49251) >127 THEN 200
300 PRINT "COUNTING"
400 CT = CT +1
500 IF PEEK(49251) >127 THEN 700
600 GOTO 400
700 PRINT CT
```

A line-by-line explanation of what the program does is given below:

100: Set the value of a counter equal to zero. This variable will be used to count the number of times we cycle through the program as a way of keeping track of time.
200: If the gate is not blocked, just repeat this step endlessly. When the gate is blocked, go on.
300: Print the word "COUNTING" to indicate that the timing cycle has started.
400: Increase the cycle counter by one.

---

[2] This measurement would let you calculate the speed of the object if you know its length. The distance that the object travels during the time it is blocking the beam is its own length. This distance divided by the measured time gives you the object's speed.

[3] This program is on the *Project Programs* disk under the name BASIC.TIMER.

500: If the gate is unblocked, leave the loop because the object must have moved out. If the gate is still blocked, continue.
600: Repeat the timing cycle, starting at line 400.
700: Print out on the screen the value of CT, which is the number of times the program went through the timing cycle.

The value of CT is a time. It is not, of course, in standard time units, but it can be converted to seconds if we know how many CT timing cycles happen every second. This is easy to measure by running the program for a long time and comparing the count displayed with the time measured with a stopwatch. It turns out to be about 72 counts per second for this program. Each count cycle represents 1/72 of a second.

This program can be used for timing an object moving through a photogate. The smallest time interval that could measured would be about 1/72 of a second. For some purposes this may be adequate, but often a "faster" timer is needed. For greater speed, the same program can be rewritten in machine language.[4] Such a program, called TIMER, is included on the *Project Programs* disk. This program will run through 3583 timing cycles per second, or about 50 times faster than the BASIC program described above. With a program that "counts" this fast, you can measure times accurately to the nearest 0.001 second. This allows us to work with objects that pass through the gate very quickly.[5]

There are several other types of timing you might want to do with a photogate. For example, you might want to measure the time it takes a pendulum to make a complete swing or how long it takes a rotating object to make one rotation. These tasks require modified versions of the timing loop. The machine-language program TIMER includes several other timing "modes." These timing modes are described below:

- **Interval Timer** - This is the kind of timing described above. The only time measured is the length of time that the photogate is blocked.

- **Pendulum Timer** - The time to be measured starts when the photogate is first blocked, continues as the gate is blocked again (half a swing), and finally stops when the gate is blocked a third time. This mode measures the time it takes a pendulum to make a complete swing.

- **Flash Timer** - This mode is the opposite of the "Interval Timer." In this case, the timing starts when the photogate is open and continues until the gate is blocked. One example of the use of this mode is in timing camera-shutter speeds. The camera body is placed across the photogate so that the gate is blocked except when the camera shutter is open. Using this mode, you can measure how long the shutter is open.

- **Strobe Timer** - In this mode, the timing starts when the photogate is first open and continues until it is open a second time. This kind of timing could be used to determine the flash rate of a strobe light or how long it takes a rotating object to make a complete rotation.

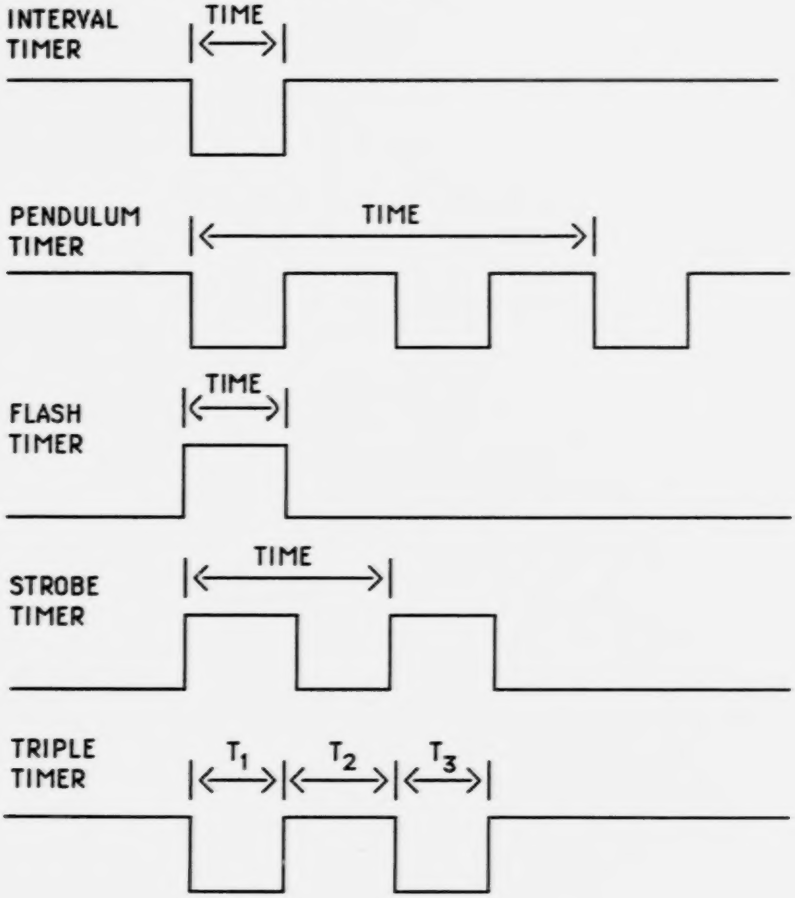
- **Triple Timer** - This mode measures three separate times. The first timing starts when the photogate is first blocked. When the gate opens, the first timing stops and the second timing begins. When the gate is blocked again, the second timing ends and the third timing begins. The third timing measures how long the photogate is blocked. All of this data allows you to do some interesting experiments, such as measuring the acceleration due to gravity of a falling object.

---

[4] Machine-language programs consist of a series of numbers that give instructions directly to the microprocessor. This kind of program is generally much more tedious to write, but tends to execute much faster than a *high-level* language such as BASIC.

[5] It is possible to have machine-language programs count much faster than the TIMER program. The TIMER program is slowed down intentionally. Machine-language loops can count at rates up to 80,000 counts/second.

The diagram below illustrates these modes for comparison. The line indicates the light level at the detector. When the line is low, the photogate is blocked; when the line is high, the gate is open. The timing interval measured is shown above each diagram.



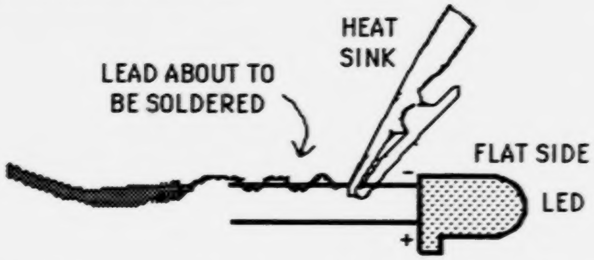
PHOTOGATE TIMING MODES

The PHOTOGATE.TIMER program on the *Project Programs* disk is a BASIC program that can be used in all of the timing modes described above. It "calls" the appropriate section of the machine–language TIMER program to do the actual timing, and then displays the counts and times. For more information on this program and how it works, refer to the "Supplemental Information" section below.

STEP-BY-STEP INSTRUCTIONS

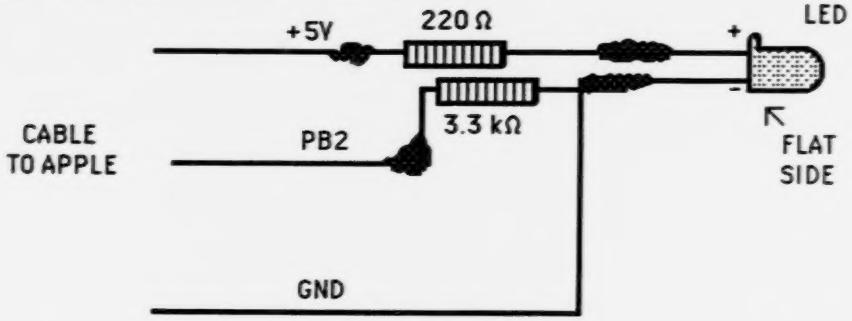
☐ 1. **Prepare the cable to the Apple game port.** See *Appendix A* for information on the type of cable you should use. Only three leads will be used on the cable: +5V, PB2 and GND. Use a fairly long cable for this project, because you will probably want to time objects at some distance from the computer. Strip the insulation from the ends of the three important wires, leaving the GND wire a little longer than the other two.

☐ 2. **Solder on the LED.** First solder a 220-Ω resistor to the +5V lead of the cable. Do not insulate this solder joint yet, because another lead will be connected here later.[6] Now identify the leads on the infrared LED. The flat side on the lip of the plastic case identifies the negative lead. Use a heat sink when soldering to the LED because it can be damaged if it gets too hot. Clip an alligator clip onto the lead you are soldering or hold the lead with a pair of pliers. Solder the negative (flat) side of the LED to the GND lead on the cable.



Do not insulate this connection now. Solder the positive side of the LED to the unsoldered end of the 220–Ω resistor.

☐ 3. **Solder the 3.3-kΩ resistor into the circuit.** Connect one side of the 3.3-kΩ resistor to the PB2 lead and the other to the GND lead as shown in the figure below. Do not insulate the PB2 connection yet; another connection will be made here later.



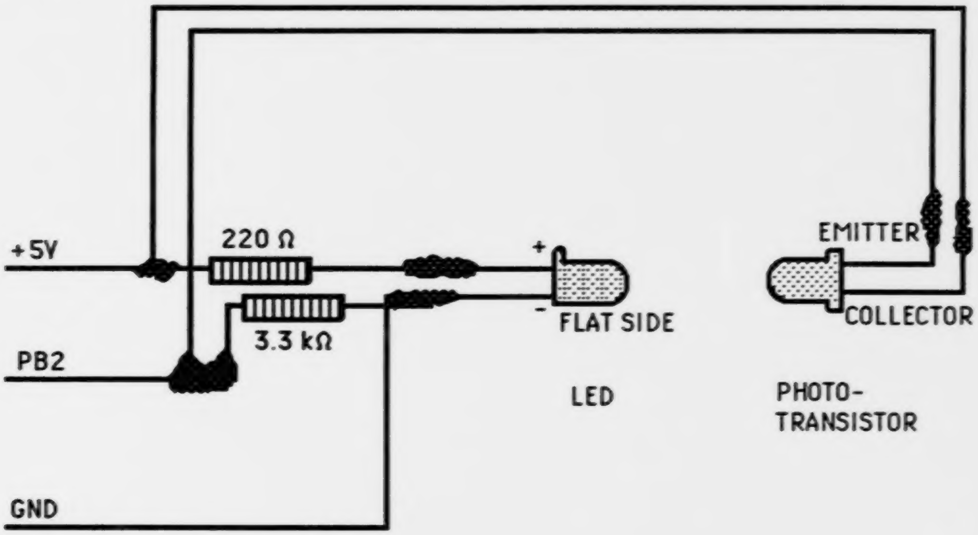
☐ 4. **Solder the phototransistor into the circuit.** First, solder a 20-cm long wire to the +5V lead from the computer at the point where the 220-Ω resistor was attached. Now, identify the leads of the infrared phototransistor. This can be confusing. If you are using an OP598 phototransistor, the "bump" on the lip of the plastic case marks the emitter lead. If you are using any other phototransistor, refer to the manufacturer's data sheets to identify the emitter and collector leads.

Using a heat sink, solder the collector lead of the phototransistor to the other end of the 20-cm wire.

Now solder another 20-cm long wire to the PB2 lead where it connects to the 3.3-kΩ resistor. Solder the other end of this wire to the emitter lead of the phototransistor. The project should now look like this:

---

[6] All of the connections in this project can be insulated with heat-shrink tubing instead of plastic electrical tape. If you do choose to use heat-shrink tubing, plan ahead—slip the tubing over the wire before making the connection.



WORKING DIAGRAM OF PHOTOGATE

Carefully insulate each solder joint and all exposed bare wire with plastic electrical tape.

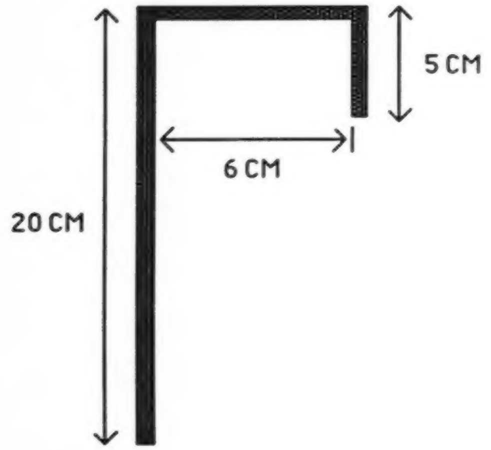
☐ 5. **Test the circuit with a meter.** Before you connect anything to the computer, it is a good idea to do a preliminary test. The phototransistor circuit can be checked quickly with a digital ohmmeter. Attach the meter to read the resistance between the +5V pin and the PB2 pin at the end of the cable. The positive lead of the ohmmeter should be on the +5V pin. With the meter connected in this way, you should notice a large change in the resistance reading when the phototransistor is exposed to bright light (a light bulb or the sun). The resistance should be several hundred thousand ohms in the dark and less than a thousand ohms in bright light.

☐ 6. **Test the circuit with the computer.** Turn off the computer and connect the photogate circuit to the computer. Turn on the computer. If the computer acts strangely, turn it off and double–check your circuit. Otherwise, run the program PB.TESTER, which is on the *Project Programs* disk. This simple program displays the status of the three PB input lines. With no light shining on the phototransistor, the status of the PB2 line should be "LOW". If you point the LED so that it is shining directly on the phototransistor, the status of the PB2 line should change to "HIGH". The PB0 and PB1 input lines are not affected by the photogate. If the photogate circuit does not work properly, refer to the "Troubleshooting" section below.[7]

☐ 7. **Attach the circuit to a solid frame.** Make a metal or wooden support with the dimensions similar to those shown below:

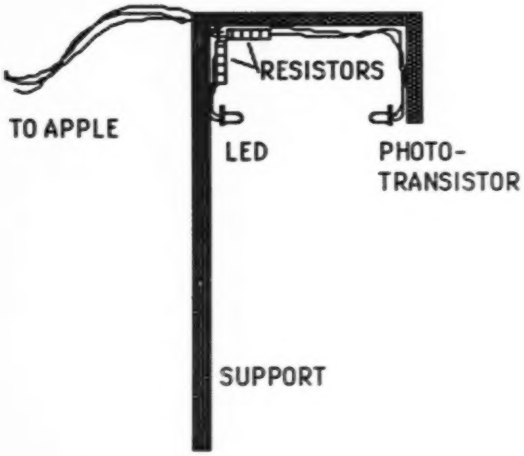
---

[7] When you hook up the circuit, do not expect to see the LED glow brightly. The infrared LED emits very little visible light. If you are in an absolutely dark room, you may see a very faint red light.



With the photogate disconnected from the computer, bend the leads on the LED and the phototransistor at right angles so that the leads can be taped to the support and the two components are pointing at each other. Tape the resistors and extra wire to the support. If the support is metal, be careful not to allow any of the exposed wire to touch it. It is essential that the LED and phototransistor be lined up properly. With the computer off, connect the photogate to the computer. Turn on the computer and run the PB.TESTER program. Align the LED and the phototransistor by bending the leads on the devices until the program indicates that the photogate is responding correctly.

The final project should look similar to this:[8]



☐ 8. **Test the final photogate timing system.** Run the PHOTOGATE.TIMER program from the *Project Programs* disk. Choose the "Interval Timer" mode from the menu. In this mode, the program should time how long the infrared light beam is blocked as an object moves through the gate. Try it by moving your finger quickly through the gate. Make sure the times seem reasonable.[9] You may notice that the photogate occasionally reports a time of zero seconds, especially when the object is moving slowly through the gate. This "false triggering" is caused by variations in the light level received at the phototransistor as the object begins to block the beam. False triggering should not occur very often when the object is moving rapidly through the photogate.

---

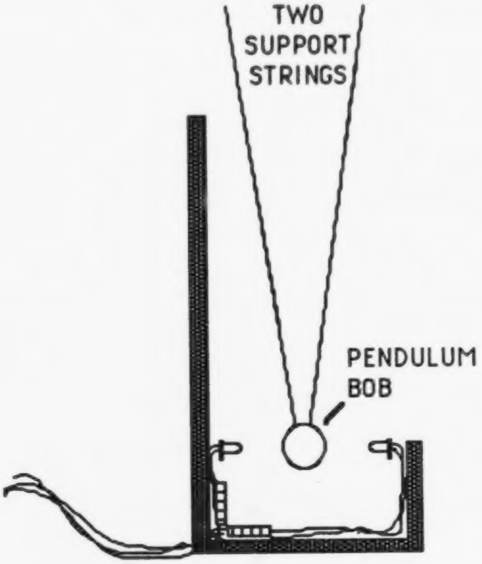
[8] There are actually many different ways of supporting the LED and phototransistor. Use a different method if you prefer. All that matters is that the two components are mounted solidly and lined up carefully.

[9] The strange symbols that appear on the screen as the timings take place indicate the status of the photogate. They will be explained in the "Supplemental Information" section.

If the photogate is not working properly, refer to the "Troubleshooting" section below.

☐ 9. **Try the other timing modes.** Make sure all four modes of the program operate as explained in the "Background Information" section above. To exit a timing mode, complete the timing cycle and then press the <Q> key.

☐ 10. **Use the "Pendulum Timer" mode to do a complete study of a pendulum.** Investigate how the period of a pendulum (the length of time for a complete swing) is affected by changes in the length, mass, and angle of swing of the pendulum. Use a pendulum with two support strings as shown below. This keeps the pendulum bob swinging in the same plane and keeps it from hitting the photogate.



Use the "Pendulum Timer" mode of the PHOTOGATE.TIMER program for this experiment.

TROUBLESHOOTING

Run the PB.TESTER program.

If the PB2 status is always "low"

- The LED and the phototransistor are not aligned—realign them.
- The LED circuit is not emitting anything—see LED tests below.
- The phototransistor circuit is not working—see phototransistor tests below.

If the PB2 status is always "high"

- The phototransistor is not working properly—see phototransistor tests below.
- The phototransistor is picking up too much room light—use a loop of black tape to block out stray light.

If the photogate response is not consistent

- If the photogate timing system works poorly with slowly moving objects, but much better with faster moving objects, there may be nothing wrong with your apparatus. An object moving slowly through the gate can cause "false triggering" as it gradually blocks the phototransistor's light. The problem is caused by minor fluctuation in the light level at the phototransistor during the transition period. The PHOTOGATE.TIMER program may erroneously report very short times. This problem may be minimized by taking the steps below or by using faster moving objects. The best solution to this problem is to add a Schmitt trigger to the photogate circuit. Such a circuit is discussed in one of the project extensions.[10]

- The LED and the phototransistor are not aligned carefully enough. Try adjusting the position of the LED or the phototransistor slightly.
- The photogate is too wide. Move the LED and phototransistor closer together.
- A different resistance value may be needed in the phototransistor circuit. Replace the 3.3-kΩ resistor with a 10-kΩ potentiometer (variable resistor). Align the LED and phototransistor carefully and then adjust the potentiometer until the gate works best.

◊ **Phototransistor test**:

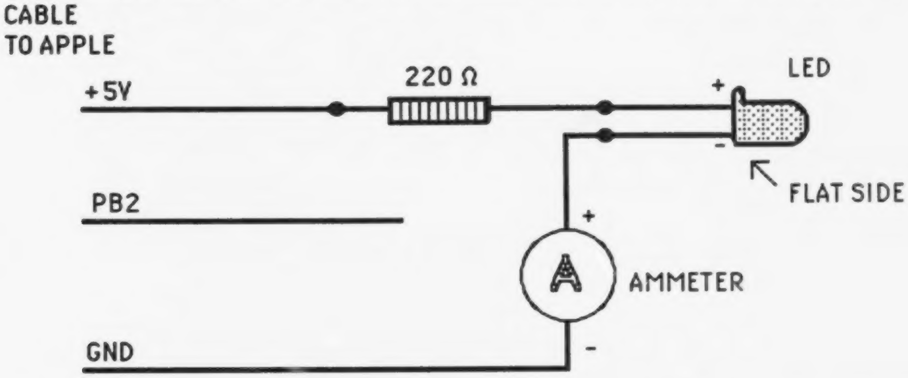
The phototransistor circuit can be checked with a digital ohmmeter. Attach the meter to read the resistance between the +5V pin and the PB2 pin at the end of the cable. The positive lead of the ohmmeter should be on the +5V pin. With the meter connected in this way, you should notice a large change in the resistance reading when the phototransistor is exposed to bright light. The resistance should be several hundred thousand ohms in the dark and less than a thousand ohms in bright light.

If this test fails, double-check the following:

- Are your phototransistor leads reversed? If you are using an OP598 phototransistor, the "bump" on the plastic case marks the emitter. If you are using a different phototransistor, check the manufacturer's data sheets to make sure you are using the leads properly.
- Are your cable connections correct? Refer to *Appendix A* and check your connections with an ohmmeter.
- Did you really use a 3.3-kΩ resistor in the phototransistor circuit?

◊ **LED test**:

There are two ways to determine if the LED is emitting any infrared radiation: (1) You can cut one of the wires leading to the LED and insert a milliammeter into the LED circuit (as shown below).



The negative lead of the meter should be attached to the GND lead. Connect the photogate to the computer and monitor the current. If the circuit is working properly, the current should be about 16 mA. If you do not get any current, or too much current, check your circuit. (2) You can check your LED by operating it in a totally dark room. If the room is very dark, you should notice a faint red glow from the LED.

---

[10] The false triggering is also reduced by slowing down the counting loop, because the slower the counting, the more unlikely it is that there would be several checks of the light level during the transition period. If you experiment with the BASIC.TIMER program, you will find that it rarely has problems with false triggering. The TIMER program was purposely slowed down to improve its performance.

If these tests indicate that the LED is not working properly, check the following:

- Are your LED leads reversed? The flattened side of the LED is the negative side.
- Are your cable connections correct? Refer to *Appendix A* and check the connections with an ohmmeter.
- Did you really use a 220-Ω resistor in the LED circuit?

PROJECT EXTENSIONS

Additional Experiments

◊ Investigate the optics of the photogate

Determine the maximum possible distance between the LED and the phototransistor that yields consistent triggering. Investigate the radiation pattern of the LED. How wide is its beam?

◊ Measure the speed of a moving object

The average speed of a moving object can be calculated by dividing the distance it travels by the time it takes to go that distance. If you know the length of a rectangular object that moves through the photogate, you can determine its speed. Run the PHOTOGATE.TIMER program and select "Interval Timer." This will measure the time that the beam is blocked as the object moves through the gate. This is how long it took the object to go its own length. The speed is the length of the object divided by this time. Using carts or air-track gliders, tape a piece of rectangular cardboard vertically to the top. Let this cardboard move through the photogate. The speed in this case is the length of the cardboard divided by the time.

◊ Measure the acceleration due to gravity

Method 1 - Using a pendulum

The period of a pendulum (the time it takes for one complete swing) depends on the pull of gravity. Physics books give the following equation for the relationship between the length of a pendulum, its period and the acceleration due to gravity.

$$T^2 = 4\,\pi^2\,(L/g)$$

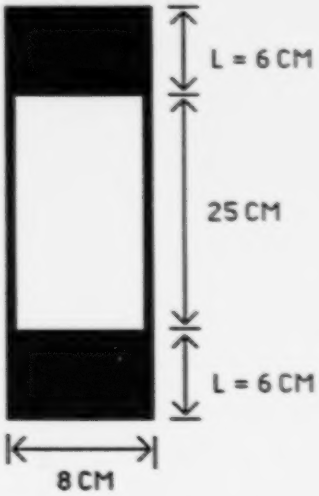
where:

π = 3.1416
L = length of pendulum measured from pivot point to the center of the pendulum bob (measured in centimeters)
T = period (time for one complete swing) in seconds
g = acceleration due to gravity in units of cm/sec^2

Carefully measure L and use the photogate and the PHOTOGATE.TIMER program to measure T. The pendulum bob should be swinging through only a small angle. Calculate g. Repeat for several different pendulums and compare your results. Also, compare your results with the accepted value for the acceleration due to gravity.

Method 2 - Using a falling object

Using the "Triple Timer" mode with a particular shape of falling object, you can also determine the acceleration due to gravity. The object to be dropped should be flat, fairly heavy and shaped as shown below:



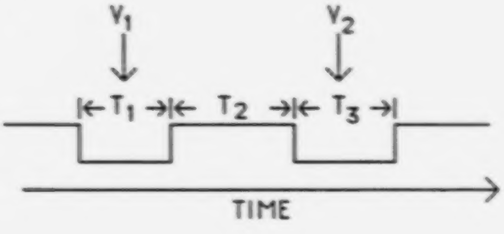
The object you make does not need to have the exact dimensions indicated. Only dimension L must be known exactly. Run the PHOTOGATE.TIMER program and select the "Triple Timer" mode. The object is held vertically and dropped through the photogate. It will interrupt the beam twice. The durations of these interruptions (T_1 and T_3) can be used to determine the speed of the object at two different moments. The "Triple Timer" mode also counts the length of time that the beam was clear (T_2). The acceleration can be determined from:[11]

$$\begin{aligned} g &= (\text{change in } V) / T \\ &= (V_2 - V_1) / (\text{time between midpoints of intervals } T_1 \text{ and } T_3) \\ &= (L/T_3 - L/T_1) / (T_2 + (T_1/2) + (T_3/2)) \end{aligned}$$

Drop the object from just above the gate.[12] Be careful about how you release the object. If the object rotates from vertical or partially misses the gate, the results are erroneous. Write down the three times and calculate g. Repeat the experiment several times. What is your best estimate of g by this method? How precise is this measurement?

---

[11] The speeds we are measuring here are average speeds while the top and bottom parts of the object move through the photogate. If the acceleration is constant, these speeds occur halfway (in time) through the gate. Therefore, the time interval during which the speed changes from V_1 to V_2 is not just T_2, but also includes half of T_1 and half of T_3.



[12] Theoretically, the point of release should have no effect on the value of g that you obtain, because g is a constant acceleration. The result should be the same no matter what height above the gate you release the object. Releasing the object from far above the gate will probably make it difficult to traverse the gate with the object still vertical.

◊ Calibration of strobe lamps

If you have access to a strobe lamp, you can measure its flash rate. For this experiment, block the light from the LED and aim the strobe light at the phototransistor. Run the PHOTOGATE.TIMER program and select the "Strobe Timer" mode. You may have to dim the room lights. The program should "count" the time between flashes.

◊ Phototachometer

You can use the "Strobe Timer" mode to determine the frequency of rotation of an object. Mount a disk with a hole in it on the shaft and place the disk in the photogate so that it blocks the gate, except when the hole rotates through. The "Strobe Timer" mode measures the time between flashes of light, which occur once each rotation. The time displayed is the period of the motion. If you know programming, modify the PHOTOGATE.TIMER program so that it displays the frequency rather than the period.

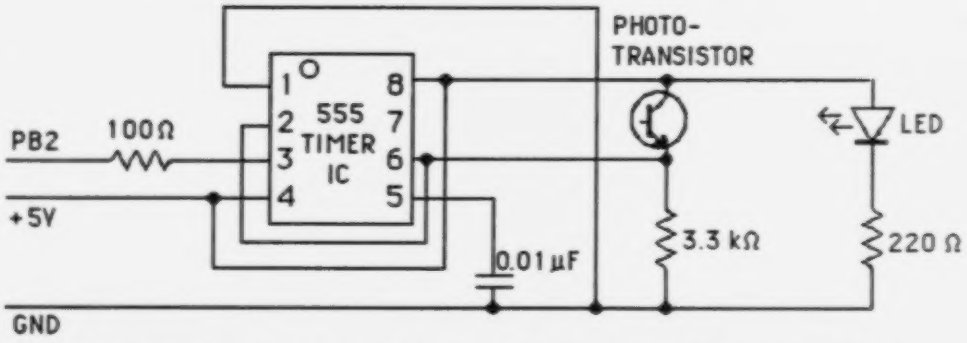
◊ Measure the duration of a flash

The photogate may be used to check the duration of the opening of a camera shutter. Place the camera (with the back removed) in the photogate so that the beam is blocked except when the shutter is open. Run the PHOTOGATE.TIMER program and select the "Flash Timer" mode. This mode can also be used to measure the duration of a flash of light from various sources such as strobe lights.

Circuit Modifications

◊ Add a Schmitt trigger to the photogate circuit

The false triggering that can occur when the simple photogate described above is used with slowly moving objects can be eliminated by adding a Schmitt trigger circuit to the photogate. A diagram for one possible circuit is shown below. Refer to an electronics text for information on how a Schmitt trigger circuit operates. This addition involves using a circuit board, one integrated circuit, a capacitor, and another resistor.



WORKING DIAGRAM OF A SCHMITT TRIGGER PHOTOGATE

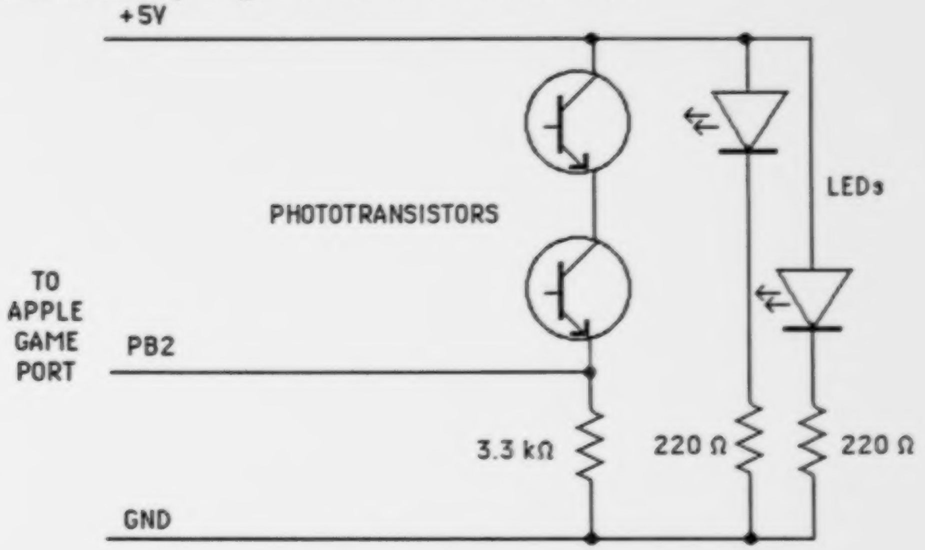
This circuit minimizes false triggering, but it also inverts the signal from the photogate; that is, it produces a low–voltage output when the simple photogate circuit produces a high signal and vice versa. Modified versions of the BASIC timing program and the TIMER program are provided on the *Project Programs* disk for use with this circuit. To do timings with the Schmitt trigger circuit, run the program SCHMITT.TIMER.

◊ Replace the LED with a laser

A phototransistor wired as described in this project can be used with a laser as a light source. Remove the LED and aim a helium-neon laser at the phototransistor. The laser produces a concentrated light, which should allow you to place it several meters away from the phototransistor. This setup gives you a larger photogate for experimentation, which can even be used to time people or cars.

◊ Using two photogates

You can add a second photogate to the circuit as shown below:



A TWO PHOTOGATE SYSTEM

With this circuit, the input line goes low if either one or both of the photogates is blocked. The "Triple Timer" mode can be used with this circuit to do more complicated timings. Set up the two photogates in a line so that a moving object (ball, cart, etc.) moves through both gates. Run the PHOTOGATE.TIMER program in the "Triple Timer" mode. As the object moves, the time it takes to go through the first gate is measured as T_1. As it continues, T_2 measures the time between the photogates. Finally, T_3 measures the time it takes to go through the second gate. The first and third times can be used to determine the speed of the object as we have done before. The sum of T_1 and T_2 is the time it took the object to move from the first photogate to the second photogate. If you measure this distance, you can determine the average speed of the object as it traveled between the two photogates from:[13]

$$\text{speed} = \text{distance} / \text{time} = \text{distance} / (T_1 + T_2)$$

For an accelerating object (of length d) you can calculate the acceleration from:[14]

$$\begin{aligned} a &= \text{change in speed} / \text{time interval} \\ &= (V_2 - V_1) / \text{time interval} \\ &= (d/T_3 - d/T_1) / (T_2 + T_1/2 + T_3/2) \end{aligned}$$

---

[13] The time it takes the front edge of the object to travel from the first photogate to the second photogate is equal to the sum of T_1 and T_2. T_1 starts when the object reaches the first gate. When the object clears the first gate, T_2 begins and then continues until the front edge of the object reaches the second gate.

[14] The speeds we are measuring here are average speeds while the object moves through each photogate. If the acceleration is constant, this speed occurs halfway through the gate. The time interval during which the speed changed from V_1 to V_2 is not just T_2, but also includes half of T_1 and half of T_3.

It is also possible to add additional photogates using the same kind of circuit. This circuit would deliver a low signal when any one of the gates was blocked. Each photogate draws about 16 milliamperes from the Apple's +5V line. Apple Computer rates this line as capable of supplying up to 100 mA. Do not exceed this limit.

◊ Build photogates for the PB0 and PB1 input lines

Using an Apple II+ computer, the simple photogate described in the "Step-by-Step Instructions" can be used to connect to either PB0 or PB1.[15] Using an Apple IIe, the simple photogate will work only if the gate is very narrow. The software to do the timing (both BASIC and machine language) will have to be modified if other switch input lines are used. Refer to the "Supplemental Information" and Appendix E for further information on how the programs operate.

Program Improvements

◊ Add the calculations of speed and acceleration to the PHOTOGATE.TIMER program and have the computer display them on the screen.

◊ Have the program save time data in memory and later display or print a data table.

◊ Add a mode to allow data to be saved on the disk.

◊ Change the PHOTOGATE.TIMER program's calibration. Line 60 of this program sets the number of counts per second at 3583. This value was determined experimentally using a particular Apple IIe. It is possible that a different computer may run at a slightly different speed. The speed of program execution is determined by the clock crystal built into the computer. Check the number of counts per second on your computer by timing over a long time interval and comparing the results with an accurate clock. Change the value of the variable CPS in line 60 if necessary.

◊ Modify the program so that it displays the times in large block digits that can be read at a distance. Use low-resolution graphics.

SUPPLEMENTAL INFORMATION

Program Design Notes

The PHOTOGATE.TIMER program is a simple BASIC program that displays the times measured in five different modes. It also includes a photogate–checking routine similar to the PB.TESTER program. When the program is run, it first sets the Apple's "top of available memory" marker at 32768 ($8000).[16] This is so the program's variables will not be stored in the section of memory used to store the machine-language TIMER program. Next, the TIMER program is loaded starting at location 32768. The main menu is displayed and the string variable M$ is used as the mode-selection indicator. The appropriate portion of the TIMER program is now called from BASIC to do the actual timing. This program also displays some timing marks on the screen. The count total is stored in the following memory locations:

36832	low byte
36836	
36840	
36844	
36848	high byte

---

[15] The differences between the switch input lines on the II+ and the IIe are discussed in "The Apple Game Port" section of the *Introduction*.

[16] The number following the "$" is in hexadecimal notation.

To interpret the counts, the BASIC program PEEKs at these memory locations and calculates the count total as follows:

$$\text{count} = \text{PEEK}(36832) + \text{PEEK}(36836) \times 256 + \text{PEEK}(36840) \times 256^2 + \text{PEEK}(36844) \times 256^3 + \text{PEEK}(36848) \times 256^4$$

This complicated system of reading the counts is required by the way the machine-language TIMER program stores the counts.[17]

The count total is now used to calculate the time. The TIMER program counts at a rate of 3583 counts per second. The time interval is calculated by dividing the counts measured by this number. The time is then rounded to the nearest 0.001 second. Both the count total and the time are now displayed on the screen. The program then checks to see if a key has been pressed, and if not, the timing is repeated. If a key has been pressed, the main menu is displayed again.

The "Triple Timer" mode is somewhat more complicated. In this mode, the TIMER program stores three count totals. They are stored as follows:

count 1	count 2	count 3	
36832	36833	36834	low byte
36836	36837	36835	
36840	36841	36842	
36844	36845	36846	
36848	36849	36850	high byte

All three of these count totals are calculated with PEEK statements. The three times are calculated and displayed. Counts are not displayed in this mode.

Timing Signals

As the machine-language TIMER program operates, you can follow the timing sequence by watching the symbols below appear on the video monitor:

: Waiting for a "BLOCKED" signal at the photogate to start timing

. Waiting for an "UNBLOCKED" signal at the photogate to start timing

- Counting until the gate is "BLOCKED"

_ Counting until the gate is "UNBLOCKED"

* Timing sequence is complete

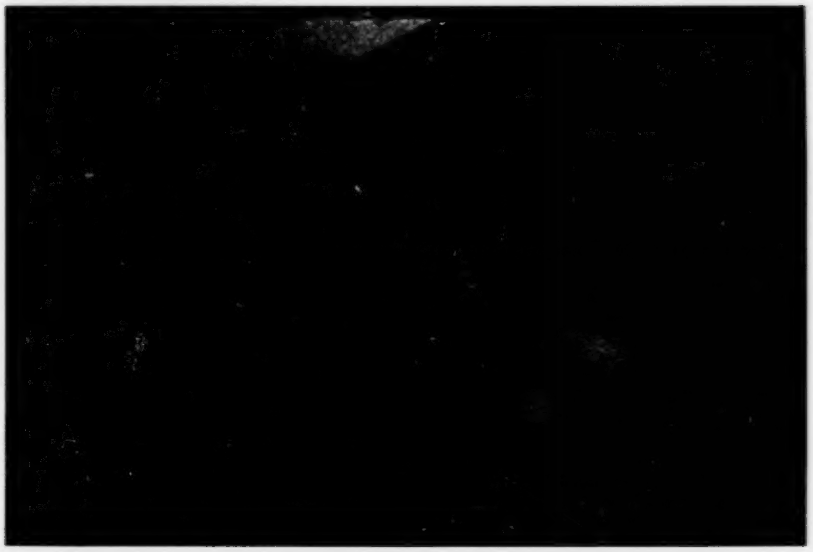
These symbols can be helpful in determining if the photogate timing system is working as you expected.

The PHOTOGATE.TIMER program was written in a straightforward manner and many REM statements are included to explain the program operation. You are encouraged to study and modify this program. *Appendix E* includes the assembly–language and machine–language program listings of the TIMER program for those interested in studying or changing this program.

---

[17] The largest number that can be stored in a single byte of memory is 255. The TIMER program adds to one location until it overflows and then marks each overflow in the next location.

PROJECT 2:
REACTION TIME



This project consists of building a simple device to measure a person's reaction time. A simulated brake pedal connected to the Apple game port will be used. Applesoft BASIC and machine-language programs are provided on the *Project Programs* disk that allow you to accurately measure the reaction times.

This project is a simple example of digital input into the computer through the game port. It is a good introductory project. The project extensions include improving the programs and doing tests on factors that affect reaction time. The "Background Information" and "Supplemental Information" sections include an introduction to machine- and assembly-language programming.

PROJECT REQUIREMENTS

Background Knowledge

- General: Some woodworking skills are needed to construct the brake-pedal simulator.
- Electronics: None
- Programming: Understanding the core project requires some familiarity with BASIC programming. Most of the project extensions involve BASIC programming. A few project extensions involve writing machine-language programs.

Parts

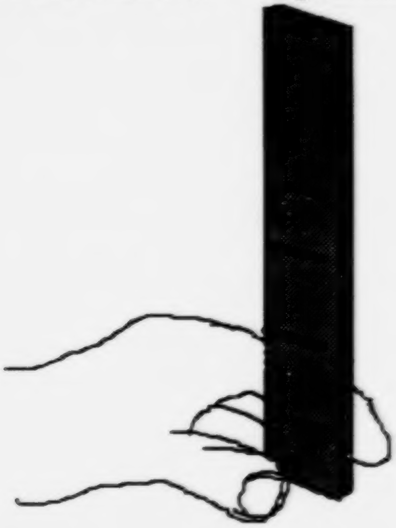
☐ cable to Apple game port (see *Appendix A*)
☐ 470-Ω resistor, 1/4 watt, 5%
☐ (3) wires to extend to the brake-pedal simulator
☐ miscellaneous scrap wood for building the brake-pedal simulator
☐ miscellaneous wood screws, bolts, etc., for building the brake–pedal simulator

Tools and Supplies

Soldering iron, solder, hammer, nails, hand saw, electric drill, drill bits and screwdriver. An ohmmeter may be useful for troubleshooting.

Background Information

There are many different ways to measure reaction times. One simple way is to hold a ruler in a vertical position with the bottom end between the opened thumb and forefinger of a friend. Instruct the friend to grasp the ruler as soon as it starts to fall. Drop the ruler.



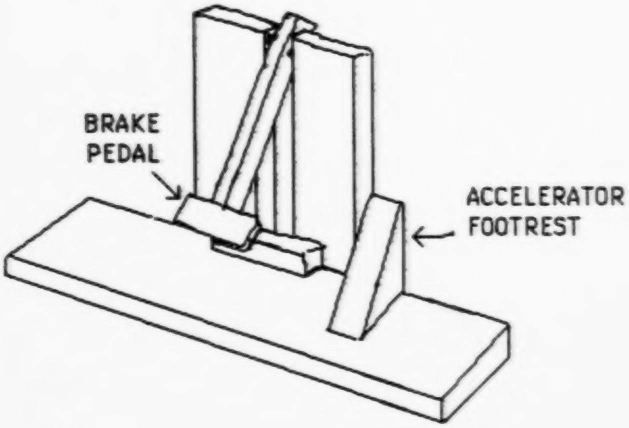
Measure how far the ruler fell before your friend was able to grab it. The physics equation below will let you calculate a reaction time (the time it took your friend to bring his or her fingers together).

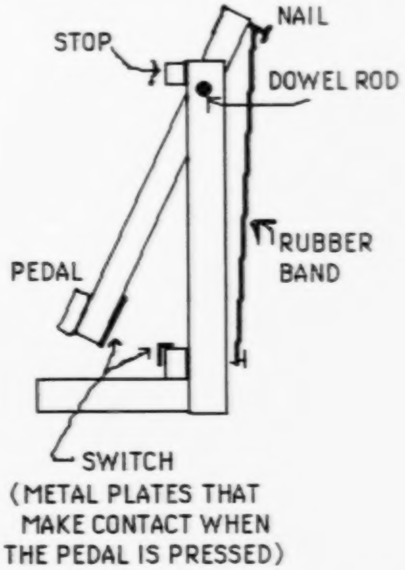
$$\text{distance fallen} = 1/2 \times (\text{acceleration due to gravity}) \times \text{time}^2$$

Substitute into this equation and use algebra to solve the equation for the time. Use either 980 cm/sec^2 or 384 inches/sec^2 for the value of acceleration due to gravity, depending on whether you are using a metric or an English ruler.

Probably the most important reaction time in today's world is the time it takes to press the brake pedal of a car in a panic–stop situation. This is the kind of reaction time we will measure in this project.

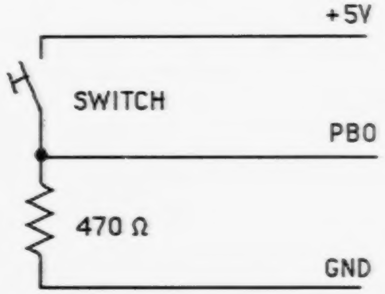
You will need to build a brake-pedal simulator—something that you can have people press in the same way they press a brake pedal in the car. The details of this brake pedal do not matter much; you can use whatever scrap material you have available. Here is one possible arrangement:


BRAKE PEDAL
ACCELERATOR FOOTREST


NAIL
STOP
DOWEL ROD
RUBBER BAND
PEDAL
SWITCH
(METAL PLATES THAT MAKE CONTACT WHEN THE PEDAL IS PRESSED)

The pedal must move so that a simple "metal–touching–metal" switch closes when the pedal is pressed all the way down.

This simple switch is connected to the Apple game port. The electrical connections to the computer are shown below:


+5V
SWITCH
PB0
470 Ω
GND

The pedal is wired so that pressing it has the same effect as pressing the push button on game paddle #0. If the pedal is pressed and the switch is closed, the PB0 line will be raised to the high voltage level. The computer can recognize this by a change in a specific memory location.[1]

The only other thing needed for our reaction-time experiments is an accurate way of measuring time intervals. There are several ways to use a computer as a timer. The simplest method is to use a BASIC program with a loop that repeats itself. The program can measure a time interval by keeping track of the number of times the program repeats the loop. A sample timing program is listed below:

```
10  J = 1 : REM SET COUNTER TO 1
20  IF PEEK(49152)>127 THEN 100 : REM CHECK FOR KEYPRESS
30  J = J + 1
40  GOTO 20
100  PRINT J : REM PRINT COUNT
```

---

[1] For information on how the computer can sense the status of the switch input lines, refer to "The Apple Game Port" section of the *Introduction*.

This program counts from the time it is RUN until a key is pressed. The count displayed is a measure of time. In order to convert this time to seconds we need to know how many loops happen per second. This can be measured experimentally by letting the timer run for a long time and comparing the count with time measured using a stopwatch. For example, the program above was run and allowed to count for 60.0 seconds. The count displayed was 4333. The program must then execute at 4333 / 60.0 or 72.2 loops per second.[2]

The limitations of this kind of a timing system are: (1) The shortest time interval that can be measured is fairly long, so the precision of time measurements is limited. In the example above, times are accurate only to the nearest 1/72 of a second. (2) The exact amount of time taken to execute each loop in BASIC is not always the same. It depends on the numbers being used for the variables. As the numbers used in the counting get larger, each loop may take slightly longer.

A better way to use the computer as a clock is to use a *machine-language program.* Machine-language programs consist of a series of two digit numbers that control the operations of the computer. Machine language is referred to as a *low-level language*, because it is written in a form that is ready for the computer to process, but it is fairly difficult for people to write or read.[3] Machine-language programs generally carry out operations much faster than BASIC programs. In the case of timing, this means that a machine-language program can "count" much faster and measure time more precisely.

The execution of a machine-language program is coordinated by an internal *clock*. This clock sends out pulses at regular intervals. The time between these pulses is called a *clock cycle*. On the Apple II series of computers, there are 1.023×10^6 of these clock cycles every second. Each machine-language program step takes a specific integer number of clock cycles.[4] This allows you to count the amount of time that any part of a program takes in terms of clock cycles. Clock cycles can be converted to seconds using the fact that there are 1.023×10^6 clock cycles per second. This calculation can be done without ever running the program.

A machine-language program that counts time by repeating a loop can count at a rate of many thousands of counts per second. The particular machine-language program we will use for this project is called TIMER.R. It is included on the *Project Programs* disk. It takes 34 clock cycles for each loop. The counting rate is then:

$$1.023 \times 10^6 \text{ cycles per second} / 34 \text{ cycles per loop} = 30{,}090 \text{ loops per second}$$

The "Supplemental Information" section later in this project lists and explains the TIMER.R program. If you are interested in learning more about machine language, refer to this section.

Fortunately, you do not have to understand machine-language programming to use the TIMER.R program. A BASIC program can load the machine-language program and then use it to measure reaction time. We will use the BASIC program REACTION.TIMER, which has a "CALL" statement to transfer control to our machine-language timing program TIMER.R. TIMER.R does the actual measurement of reaction time by counting timing loops. Then the REACTION.TIMER program takes over again and displays the reaction time. The essential parts of the REACTION.TIMER program are listed and discussed below.

---

[2] You might want to check this on your computer.

[3] *High-level languages* such as BASIC, Pascal, Logo and FORTRAN are written in a form that is easier for people to work with, but these languages must be translated into a different form before the computer can process the instructions. Some high-level languages are converted to a form that the computer can process one line at a time as the program runs. This is referred to as an *interpreted* programming language. Applesoft BASIC is an example of this type of programming language. Interpreted programming languages are generally slow, because of the line-by-line translation.

[4] Any book on 6502 machine language will include a table of the number of clock cycles taken for each operation. One good example is *Programming and Interfacing the 6502* by Marvin L. DeJong.

```
45 PRINT D$; "BLOAD TIMER.R, A$300" : REM LOAD MACHINE-LANGUAGE TIMING
PROGRAM
60 R = RND(1) * 10000 : REM SELECT A RANDOM NUMBER BETWEEN 0 AND 10000
65 PRINT "GET READY"
70 FOR J = 1 TO R
80 NEXT J
90 PRINT "            STOP"
100 CALL 768: REM CALL MACHINE-LANGUAGE PROGRAM TO COUNT TIME
120 C = PEEK(973) + 256 * PEEK (974) + 65536 * PEEK(975): REM CALCULATE COUNT
140 T = C *34 / 1.023E6: REM CONVERT COUNTS TO TIME
160 PRINT "REACTION TIME:  "; T ; " SECONDS"
```

Line 45 loads the machine-language timing program into memory. Lines 60-80 warn the person being tested to be ready and then cause the program to stall for a random time interval before anything happens. Line 90 displays the "STOP" signal, which tells the person being tested that it is time to press the brake pedal. Line 100 transfers control to the machine-language program TIMER.R, which is in memory starting at address 768. This program repeats a timing loop until the brake pedal is pressed. The BASIC program then takes control again. and checks the counts that were measured by the machine-language program. Line 120 does this by reading the values that are stored in three memory locations.[5] Line 140 converts the count to a time in seconds. Finally, line 160 displays the reaction time.

The combination of the brake-pedal apparatus and the programs on the *Project Programs* disk will allow you to measure reaction time accurately. The person being tested should sit in front of the brake–pedal simulator in a normal position for driving. The REACTION.TIMER program will display a "GET READY" message on the video monitor. It then will wait a random time period (up to about 10 seconds) and then display "STOP". At this point, the person being tested should press the brake pedal as quickly as possible. The reaction time in seconds is displayed on the screen. The testing cycle then repeats.

The project extensions involve improving on this simple testing procedure and doing experiments on factors that affect reaction time.

Note to Apple IIe Users: The brake-pedal simulator is electrically connected in the same way as the <OPEN APPLE> key on an Apple IIe. You can use the <OPEN APPLE> key as a trial "brake pedal" to see how the REACTION.TIMER program works.[6] You can also use the <OPEN APPLE> key to test your eye-hand reaction time.

STEP-BY-STEP INSTRUCTIONS

☐ 1. **Assemble a brake–pedal apparatus.** Use the illustrations in the "Background Information" section above as examples, but modify the design as you see fit to use the scrap wood or metal you have available. The essential considerations are:

- There must be a movable pedal that approximately simulates a brake pedal.
- When the pedal is pushed down, two pieces of metal must come into contact to act like a switch.
- When the foot is taken off the pedal, the pedal should move up, opening the switch.

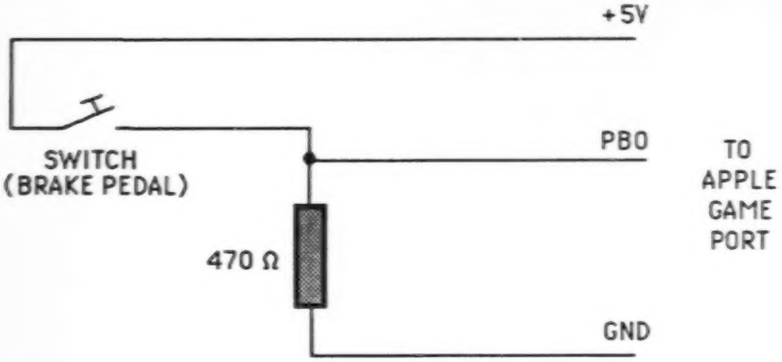
---

[5] Refer to the "Supplemental Information" section for details on how this count is stored and interpreted.

[6] It is not a good idea to use your foot on this "brake pedal."

- Include an extra block of wood to act as an accelerator pedal. This is really just a place for your foot to rest (as it would while driving) as you wait for the reaction test.
- The brake pedal and the accelerator "footrest" should approximate a normal driving situation when the "driver" sits in a chair in front of them.
- The whole apparatus should be fairly rugged, since people are not very gentle when using their feet.

☐ 2. **Attach the game-port cable.** Refer to *Appendix A* for suggestions on the cable to use and for tips on identifying the leads. For this project you will use the +5V, PB0 and GND leads. Strip a centimeter or so of insulation from each of these leads. Complete the circuit as shown in the diagram below:



WORKING DIAGRAM OF BRAKE-PEDAL SIMULATOR CIRCUIT

Solder one end of the 470-Ω resistor and a section of wire to one side of the switch on your brake-pedal simulator.[7] Connect the other end of the wire to the PB0 lead on the game-port cable. Solder a wire from the other end of the resistor to the GND lead of the game-port cable. Finally, connect the other side of the switch to the +5V lead of the cable.

☐ 3. **Test the brake-pedal apparatus.** Turn off the computer. Connect the brake–pedal apparatus to the computer's game port. Turn on the computer and run the PB.TESTER program. The monitor will display the status of the three PB input lines. The output should read "PB0 LOW" when the pedal is up (not pressed). It should display "PB0 HIGH" when the pedal is pressed. The other two PB lines are not affected. If the circuit does not work properly, refer to the "Troubleshooting" section below.

☐ 4. **Try out the REACTION.TIMER program.** Run the program REACTION.TIMER that is on the *Project Programs* disk. Sit on a chair in front of the brake-pedal simulator. This program will delay for a random time interval (up to 10 seconds) and then display the word "STOP". When "STOP" appears, you should press the brake pedal as quickly as you can. The program should display your reaction time.[8] The program then repeats. To exit this program, press <CONTROL> <C>.

☐ 5. **Try measuring the reaction time of your friends.** Run the reaction time test a number of times with several different people. Require that all of the test subjects keep the right foot on the "accelerator pedal" foot rest while waiting for the "STOP" signal. This helps standardize the experiment. Record all results and calculate average reaction times for each person.

---

[7] If you are using large pieces of metal to make up the switch, you may have a difficult time soldering to it. If you cannot solder this joint, just make good mechanical contact between the resistor, the wire, and the metal of the switch.

[8] Notice that the time is displayed with far too many digits to be meaningful. A good program should round off the values displayed to a reasonable number of significant digits. This is one of the recommended project extensions.

TROUBLESHOOTING

If the brake–pedal apparatus does not work properly:

◊ Double-check the leads on the cable from the game port on the computer to the brake pedal. Are you sure you are using the correct leads for +5V, GND and PB0 (not PB1 or PB2)? The best way to test this is to use an ohmmeter to check for conductivity between the pin on the connector that goes to the computer and the wires at the pedal. The pins on the game–port connector are identified in *Appendix A*.

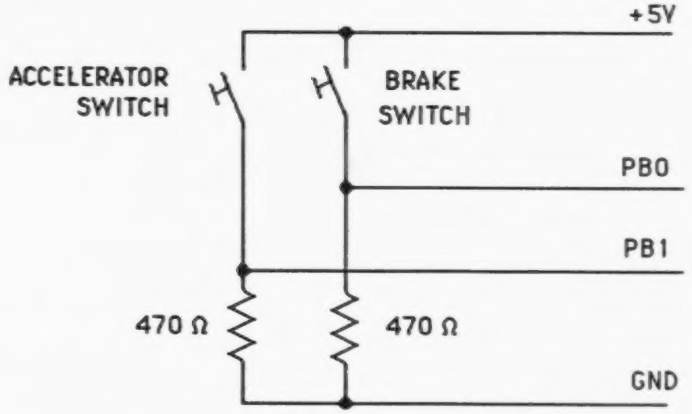
◊ Make sure that the switch is really acting like a switch. The metal pieces that make up the switch should touch together when the pedal is pressed all of the way down and the circuit should be open when the pedal is up. If necessary, check this with an ohmmeter.

PROJECT EXTENSIONS

Circuit Modifications

◊ One way to cheat on the reaction timing test is to move your foot close to the brake pedal while you are waiting for the "STOP" signal. You can eliminate this by adding an accelerator pedal that includes a switch. With this addition and a slight change in the program, the test subject will be forced to keep the accelerator pressed until the "STOP" signal is given.

Build an accelerator pedal that functions in the same way as the brake-pedal simulator. Two metal contacts should touch together only when the accelerator pedal is pushed. Make it look and feel like an accelerator pedal if possible. The electrical connections needed are:



BRAKE- AND ACCELERATOR- SIMULATOR CIRCUIT

To use this equipment, you will need a slightly more complicated version of the BASIC program. The program R.TIMER.WT.ACC is included on the *Project Programs* disk specifically for this project extension. This program checks to make sure the accelerator pedal is held down before giving the "STOP" signal.

Note to Apple IIe Users: The <CLOSED APPLE> key on an Apple IIe functions just like the proposed accelerator pedal discussed above. You can use the <OPEN APPLE> key as a brake–pedal simulator and the <CLOSED APPLE> key as an accelerator-pedal simulator to see how the R.TIMER.WT.ACC program works.

◊ Change to a different type of switch. Digital input into the computer can be accomplished by any type of switch. Replace the homemade switch used in the brake-pedal apparatus with a different switch. In fact, any electronic circuit that changes its voltage output from near 0 to above 2 volts can be used for the input signal.

◊ Build another switch that attaches to PB2. Use a circuit similar to the one used in the brake–pedal simulator. To use this input signal, you will modify both the BASIC and the machine-language programs.

Program Improvements

The reaction-timing programs included on the disk are very simple. There are many changes you can make to improve them:

◊ Have the program round off the reaction times to a reasonable number of significant figures.
◊ Add distractions on the screen as the test subject waits for the "STOP" signal. Have other words, color patterns or lines drawn on the screen as the subject waits for the signal. This should help simulate the distractions of driving.
◊ Add a method of keeping track of a person's results. A good system should record all values, the minimum, the maximum and the average. Perhaps there should be some provision for deleting faulty timings, such as when a person was not ready.
◊ Add a system to keep track of the results of several individuals and compare them. Perhaps the results should be saved on disk or made into a bar graph or a histogram. The data and graphs can be printed on a printer.
◊ Have the computer calculate the distance that a car traveling at 55 mi/hr would travel in the reaction time measured. This could be displayed right along with the reaction time after each test.
◊ If you have added a working accelerator pedal and are familiar with machine-language, change the program so that two different times are measured. Change the machine-language program TIMER.R so that the time from the "STOP" signal to the release of the accelerator pedal is first measured. Then have the program measure the time from the release of the accelerator to the pressing of the brake pedal. Refer to the "Supplemental Information" section below and *Appendix E.*
◊ Improve the appearance of the program. Clear the screen before each test. Have the "stop" indicator appear in the shape of a stop sign. Have the reaction time displayed in large digits so that it can be read at a distance.
◊ This project is a very simple example of digital ("on or off") input into the computer. You can probably think of other situations where it would be useful for the computer to be able to sense things about the "outside world." Write your own programs that check and respond to the status of the PB input lines. One example is an alarm system that causes the Apple computer's speaker to make noise when someone moves a door and opens a switch.

Additional Experiments

◊ Investigate reaction times among different groups of people. Do younger people tend to have faster reaction times? How about experienced drivers versus nondrivers? Is there a significant difference in reaction time between men and women?
◊ Investigate factors that can cause changes in reaction times for a particular individual. Systematically investigate how each of the following affects reaction time:

- distractions
- background music
- fatigue
- caffeine
- time of day

SUPPLEMENTAL INFORMATION

Additional Information on the Machine-Language Program

This section is intended to give a brief introduction to machine- and assembly-language programming. It is not intended to teach you machine language, but it hopefully will give you an idea of what machine- and assembly-language programs are like. If you are interested in learning more about this kind of programming, refer to one of the books on the subject listed in *Appendix C*.

The machine-language program TIMER.R is listed below:

```
$A9 $00 $8D $CD $03 $8D $CE $03 $8D $CF $03 $2C $61 $C0 $30 $0F $EE $CD $03
$D0 $0D $EE $CE $03 $D0 $0C $EE $CF $03 $D0 $0B $60 $F0 $FD $EA $EA $EA
$EA $EA $EA $EA $EA $4C $0B $03
```

The two-digit numbers are written in hexadecimal (base 16) notation.[9] This machine-language program would be difficult for people to write or interpret directly. For this reason, machine language is usually written and examined by using its *assembly-language* equivalent. Assembly language is machine language in a form that is slightly easier for people to understand. The process of converting an assembly-language program to machine language is called *assembly*. The reverse process is called *disassembly*. Most Apple II computers have built-in facilities for doing both of these conversions.[10]

The list below shows the TIMER.R program with the machine language on the left and the assembly-language equivalent on the right.[11] The hexadecimal numbers preceding the colons are the memory addresses where the program is stored.

Memory Location	Machine Language	Assembly Language	# of Clock Cycles[12]
$0300:	$A9 $00	LDA #$00	2
$0302:	$8D $CD $03	STA $03CD	4
$0305:	$8D $CE $03	STA $03CE	4
$0308:	$8D $CF $03	STA $03CF	4
$030B:	$2C $61 $C0	BIT $C061	4
$030E:	$30 $0F	BMI $031F	2+
$0310:	$EE $CD $03	INC $03CD	6
$0313:	$D0 $0D	BNE $0322	2+
$0315:	$EE $CE $03	INC $03CE	6
$0318:	$D0 $0C	BNE $0326	2+
$031A:	$EE $CF $03	INC $03CF	6
$031D:	$D0 $0B	BNE $032A	2+

---

[9] The number system you are most familiar with is *decimal* or base 10. Counting in the hexadecimal number system goes: 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12, 13, 14, . . . The largest two-digit hexadecimal number is FF, which is equivalent to 255 in decimal notation. All Apple machine-language programs are in machine language. Hexadecimal numbers in this book are written preceded by a dollar sign to distinguish them from decimal numbers.

[10] All Apple II computers can do disassembly. This feature is built into the Apple's ROM memory. Most Apple II's include a mini-assembler that allows you to write and assemble simple assembly-language programs. Refer to your Apple reference manuals for more information.

[11] You can produce a listing very similar to this on your own computer by entering the following:

```
BLOAD TIMER.R
CALL -151
300L
```

[12] This is the number of 6502 microprocessor clock cycles required to execute each operation. The symbol "2+" in the table indicates an operation that normally takes two clock cycles, but takes three clock cycles whenever it causes branching to another part of the program. By studying how the program proceeds and by counting the clock cycles, you can determine how long each timing loop of the program takes.

```
$031F:     $60            RTS           6
$0320:     $F0 $FD        BEQ $031F     2+
$0322:     $EA            NOP           2
$0323:     $EA            NOP           2
$0324:     $EA            NOP           2
$0325:     $EA            NOP           2
$0326:     $EA            NOP           2
$0327:     $EA            NOP           2
$0328:     $EA            NOP           2
$0329:     $EA            NOP           2
$032A:     $4C $0B $03    JMP $030B     3
```

The table below explains what each of the lines of the program does:

Memory Location	What the Machine-Language Instruction Does
\$0300- \$0301	Loads a zero into a special storage location called register A.
\$0302- \$030A	Stores the value of register A to the three memory addresses that will be used to store the count.
\$030B- \$030D	Checks the status of the PB0 input (the brake pedal).
\$030E- \$030F	If the PB0 line is high (pedal pressed), control transfers to \$031F so that counting ends.
\$0310- \$031E	Records a count each time through. Memory location \$03CD is increased by one each time these instructions are executed. This memory location (and all others) can hold an integer up to 255 (\$FF). When it reaches 255, the next time through it overflows and goes to zero, but the next memory location (\$03CE) is increased by one. Each number in this memory location represents 256 counts. In a similar way, if the second memory location overflows, it returns to zero and the third memory location (\$03CF) increases by one. Each number in the third memory location represents 256 x 256 or 65,536 counts.
\$031F	Return to the BASIC program (used when the machine-language program is completed).
\$0322- \$0329	These are "no operation" steps; they do nothing except take up some time. They take two clock cycles per instruction. They are used to make the program take exactly the same number of clock cycles (34) regardless of whether one of the counters overflows or not.
\$032A- \$032C	Transfers control back to the instruction at \$030B and restarts the loop.

Memory locations $030B through $032C are the main loop that is repeated over and over. Each time through this loop will take 34 clock cycles.[13] Since there are 1.023×10^6 clock cycles per second on an Apple computer,

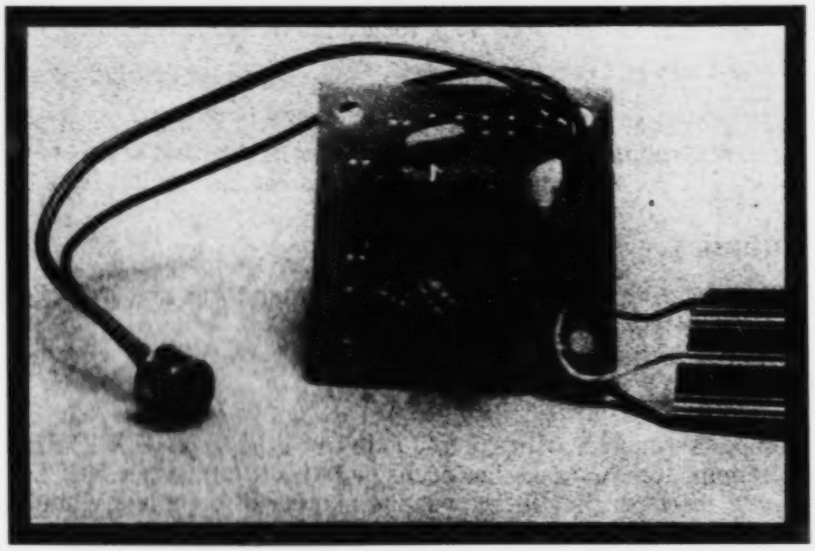
$$\text{time per loop} = 34 \text{ clock cycles} / 1.023 \times 10^6 \text{ clock cycles per sec}$$
$$= 3.324 \times 10^{-5} \text{ seconds}$$

The BASIC program REACTION.TIMER uses the count number stored in the three memory locations ($3CD, $3CE and $3CF) and the number of loops per second to calculate the reaction time in seconds.

---

[13] You might want to try to step through this program as it executes, counting clock cycles to see if you can confirm that it does take 34 cycles, regardless of the status of the count registers.

PROJECT 3: MICROPHONE/AMPLIFIER



This project will let you use the Apple II+ or IIe as a frequency meter to study sound waves. You will build a simple microphone/amplifier circuit that connects to the Apple game port. This circuit, together with programs on the *Project Programs* disk, allows the computer to determine the frequency of sound from various musical instruments. A fairly "pure" sound wave is required; good sources include tuning forks, flutes, recorders, and audio oscillators. The project extensions include experiments with the physics of music and improvements to the BASIC program.

PROJECT REQUIREMENTS

Background Knowledge

- Electronics: This project requires assembly of a circuit containing about a dozen components. Detailed instructions are given for the placement of the components on the circuit board. Previous experience with soldering is recommended.
- Programming: No programming experience is required for the core project. Some of the project extensions require experience with BASIC programming.

Parts

- ☐ electret microphone (RS #270-092)
- ☐ LM3900 Quad Norton Op Amp (These ICs are available from most electronic-parts suppliers. They are sometimes available at Radio Shack as #276-1713.)
- ☐ 14-pin IC socket (RS #276-1999)
- ☐ cable to Apple game port (see *Appendix A*)
- ☐ circuit board (The "Step-by-Step Instructions" assume you are using one-half of a Radio Shack #276-159 circuit board. If you use a different circuit board, the project will be more difficult.)
- ☐(4) 10-MΩ resistors, 1/4 watt, carbon composition, 5%

☐ 2.2-MΩ resistor, 1/4 watt, carbon composition, 5%
☐ 1-MΩ resistor, 1/4 watt, carbon composition, 5%
☐ 2.2-kΩ resistor, 1/4 watt, carbon composition, 5%
☐ 50-kΩ potentiometer, single turn, PC board mount (RS #271-219)
☐(2) 0.1-µF capacitors,10 volt, ceramic or metal film (RS #272-135 or RS #272-1053)
☐(2) short jumper wires

Note: A "Microphone/Amplifier Parts Kit" containing all of the parts listed above is available from Vernier Software for $12.00. An order form is included in the back of this book.

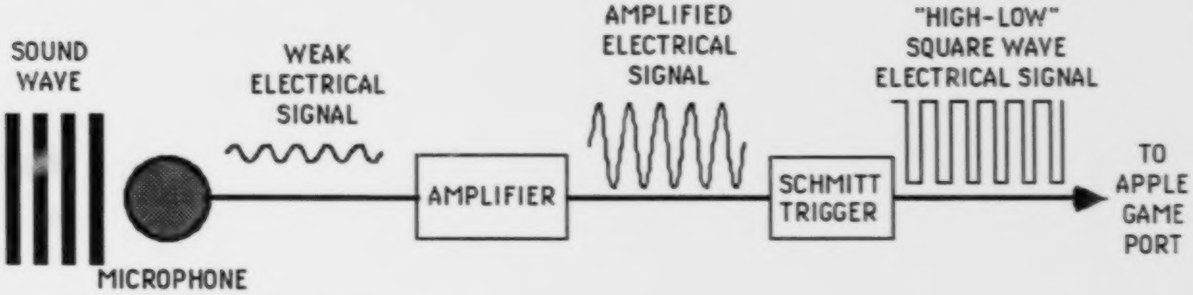
Tools and Supplies

Solder, soldering iron, wire cutters and wire strippers. A multimeter and an oscilloscope can be used for troubleshooting the project.

BACKGROUND INFORMATION

One of the most important characteristics of a sound wave is its *frequency*. The frequency of a sound is the number of complete vibrations that occur in a specific time period. The normal units for frequency are *hertz*; the frequency in hertz is the number of complete vibrations in one second. Frequency is closely related to, but not exactly the same as, the musical term pitch. The frequencies of sound waves that humans can hear range from about 20 Hz to about 15,000 or 20,000 Hz.[1]

The circuit used in this project can detect sound waves and produce an electrical signal that the Apple computer can analyze. The conversion process is outlined below. The graphs illustrate how the voltage varies with time at each step.



There are three principal parts to the circuit:

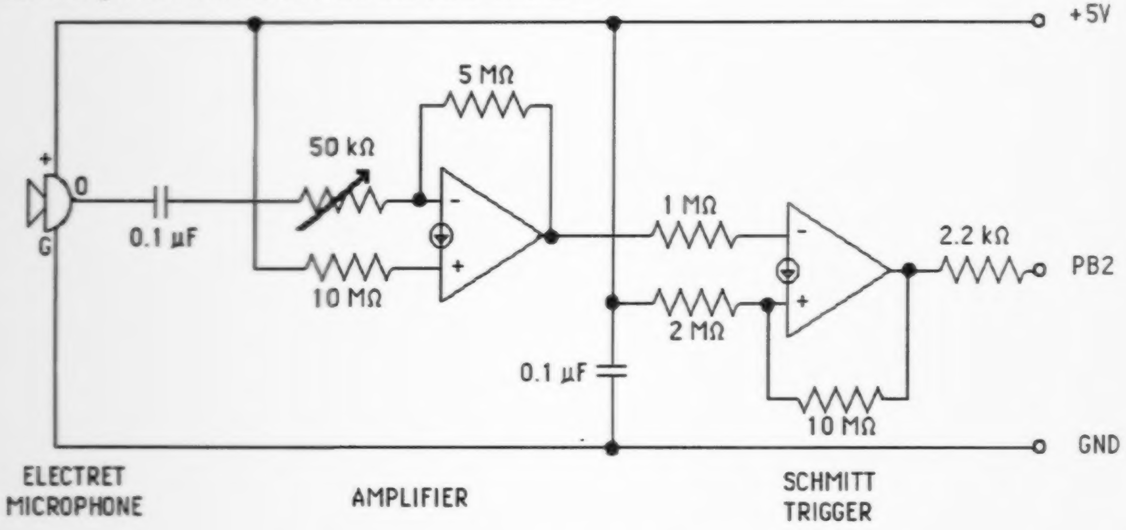
1. An electret microphone responds to air pressure changes caused by the sound wave and produces a small electrical current. This current varies with the sound wave.
2. An amplifier increases the strength of the electrical signal.
3. A *Schmitt trigger* is used to produce a *square wave* that has the same frequency as the original sound wave. The square wave signal has only two possible values: low (near 0 volts) and high (near 5 volts). This signal is fed into the game-port PB2 lead.

The electronic circuit needed to do this conversion process can be built fairly easily and inexpensively. The microphone is available for less than $3.00 from Radio Shack stores. The rest of the circuit can be built using a single integrated circuit and a few resistors and capacitors. The IC is called an *LM3900 Quad Norton Operational Amplifier*. The term "quad" is used to indicate that this IC actually contains four separate operational-amplifier circuits.[2] Operational amplifiers are common "building blocks" of modern electronics. We will use two of the four operational amplifiers. One operational amplifier is connected to act as the amplifier for the circuit. A second operational amplifier is wired to act as the Schmitt trigger, which we need to get the signal ready for the computer.[3]

---

[1] The upper frequency limit of your hearing depends on your age, health and other factors, such as how often you have been exposed to loud noises.

The complete schematic of the circuit is shown below:



MICROPHONE/AMPLIFIER SCHEMATIC

This circuit produces a voltage signal on the PB2 line of the game port that varies from high to low, with the same frequency as the original sound wave. The 50-kΩ potentiometer acts as a *gain control.* It controls the sensitivity of the microphone.

The computer can now be used to measure the frequency of the voltage signal entering the game port. This is done by using programs on the *Project Programs* disk. The program FREQUENCY is a short BASIC program that displays the frequency being received on the PB2 line. When you run the FREQUENCY program, it transfers control (using a CALL statement) to a machine-language program named FREQ. This program continuously checks the status of the PB2 input line. Each time the voltage on this line changes from low to high, a count is registered. While the program is accumulating counts, it also keeps track of the time it has been counting. It stops counting when it has monitored the PB2 line for exactly 0.501563 seconds. At this point, the FREQUENCY program takes over again. It determines the number of counts that have been registered and divides the number of counts by the time interval to determine the frequency:

frequency = counts / time interval = counts / 0.501563 seconds

The number of counts accumulated and the frequency are displayed. The frequency is rounded to the nearest hertz.[4] The FREQUENCY program is very simple. Many possible improvements are left as project extensions.

---

[2] The LM3900 is manufactured by National Semiconductor Corporation. This operational amplifier is a little different from most other op amps in that it senses the current instead of the voltage at its input leads . For information on this IC, refer to National Semiconductor's *Linear Databook* or *Electronic Design with Off-the-Shelf Integrated Circuits* by Z. H. Meiksin and Philip C. Thackray.

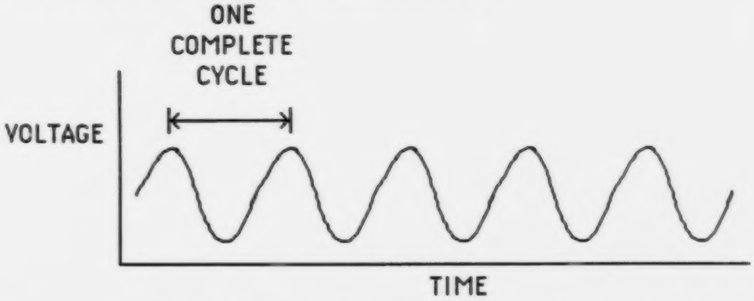
[3] For information on how a Schmitt trigger circuit operates, refer to an electronics text such as *The Art of Electronics* by Horowitz and Hill.

[4] Since the count is always an integer and the time interval is about 1/2 second, the smallest change in frequency that can be measured is 2 hertz. This is referred to as the *sensitivity* of the system.

There are two limitations on this frequency-measurement system that should be kept in mind when using it:

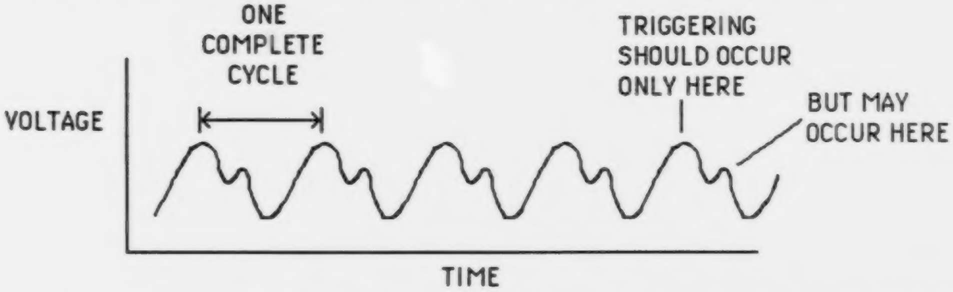
1. The FREQ program counts pulses received over a 0.501563-second time interval. If the sound starts or ends in the middle of a particular time interval, as it almost always does, that reading will be erroneous. This means that as a note is played, the first and last frequency readings taken are usually not meaningful. It also means that you need to hold a steady tone for longer than one second to get meaningful frequency measurements.

2. For good results, a fairly "pure" tone is required; that is, the sound wave being analyzed should not contain a lot of intense *overtones*. Overtones are higher frequency wave components that usually accompany the *fundamental* note.[5] If the overtones in the sound have enough intensity, they can cause false readings. When a sound containing no overtones is being played, the output of the microphone looks like a sine wave as shown below:



The amplifier and the Schmitt trigger circuit "trigger" on the peak voltages and produce a square wave output with the same frequency as the sine wave. The correct frequency is displayed by the computer.

When a sound wave contains overtones, the shape of the voltage versus time graph for the output of the microphone is changed to a more complex shape. An example is shown below:



If the overtones are intense enough, the extra variation in voltage can trigger the Schmitt trigger circuit. More than one count may be registered per cycle. In the example above, it is possible that the extra "bump" on the voltage versus time graph may result in extra triggering of the circuit on some cycles. As a result, the output of the Schmitt trigger may have a higher frequency than the original wave. The computerized frequency-measurement system may then report a higher frequency than it should. You may even get consistent, but incorrect results in some cases.

Instruments that usually work well with the microphone/amplifier circuit include: human whistles, recorders, flutes, and tuning forks. Stringed instruments that are plucked, such as guitars, do not work very well in most cases, because their sound intensity decreases quickly. Human singing voices do not work well because they often contain many strong overtones. Some musical instruments may also produce overtones too intense for good frequency readings.

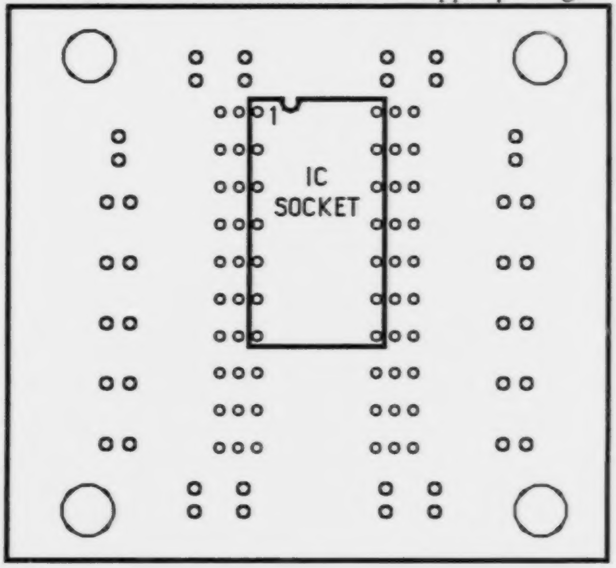
---

[5] Any book on the physics of music will discuss overtones, fundamentals and tone quality. Two examples are: *Physics and the Sound of Music* by John S. Rigden, John Wiley and Sons, New York, 1977 and *Horns, Strings and Harmony* by Arthur H. Benade, Anchor Books, Garden City, NY, 1960.

STEP-BY-STEP INSTRUCTIONS

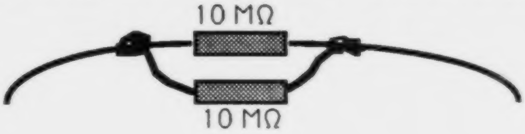
The instructions and diagrams below assume you are using one-half of a Radio Shack Dual IC Circuit Board (RS #276-159) for assembling this project. If you are using any other circuit board or a prototyping board, the component placement will be left to you. Use the schematic diagram of the microphone/amplifier circuit in the "Background Information" section above as a guide for assembling this project.

☐ 1. **Solder the IC socket to the circuit board.** Hold the circuit board with the words "RADIO SHACK" at the bottom of the circuit board on the side of the board away from you. Place the IC socket on the side of the circuit board without copper plating, as shown in the diagram below:



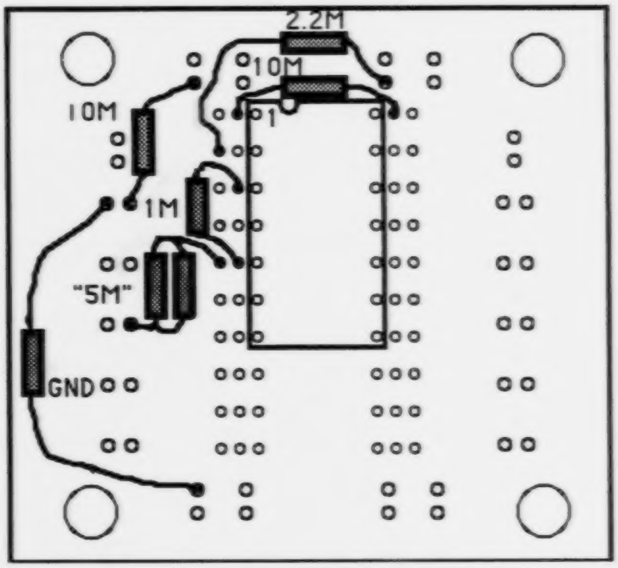
Pin #1 of this socket should be positioned at the top left-hand corner, as shown in the diagram. It is important that the socket be placed exactly as shown in the diagram. Solder all 14 of the pins to the copper-plated side of the circuit board. Do not plug the LM3900 IC chip into the socket until you are completely finished with the project.

☐ 2. **Make a 5-MΩ resistor by combining two 10-MΩ resistors.** 5-MΩ resistors are not readily available, so we will make one by combining two 10-MΩ resistors in parallel. Solder the two 10-MΩ resistors side by side and cut off the excess lead wire on one of the resistors.



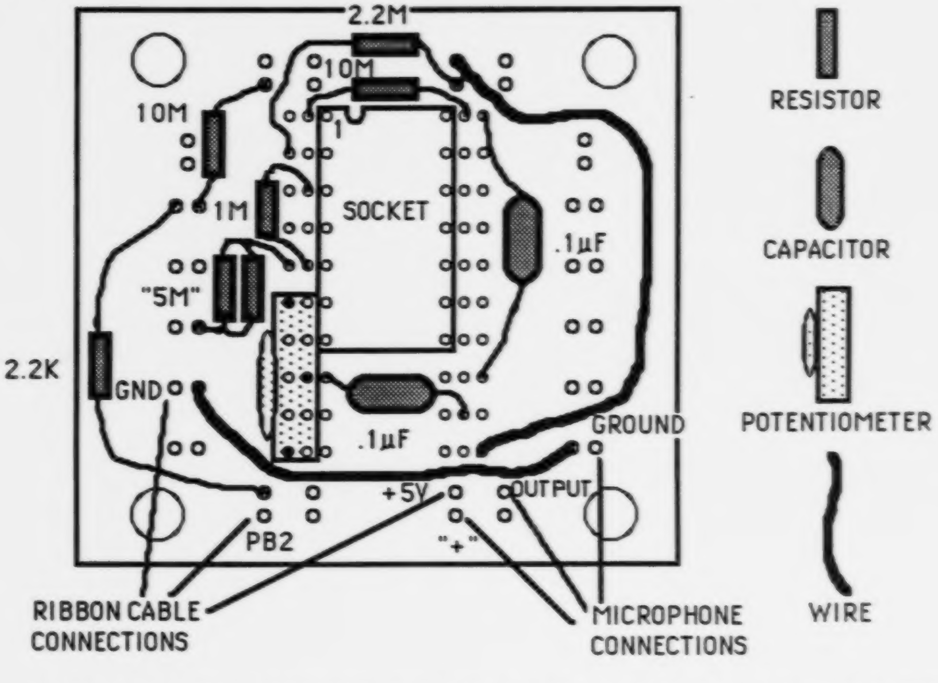
5-MΩ RESISTOR

☐ 3. **Solder all of the resistors onto the circuit board.** The diagram below shows the correct locations for all 6 resistors needed (treating the two 10-MΩ resistors soldered together as one).


2.2M
10M
10M
1
1M
"5M"
2.2K
GND

MICROPHONE/AMPLIFIER CIRCUIT (TOP, COMPONENT SIDE)

☐ 4. **Solder the two capacitors, two jumper wires and the 50-kΩ potentiometer to the circuit board.** Use the diagram below to indicate the proper holes to use for the leads.[6] You may need to ream out the holes on the circuit board for the potentiometer leads.


2.2M
10M
10M
1
SOCKET
1M
.1µF
"5M"
2.2K
GND
.1µF
GROUND
+5V
OUTPUT
PB2
"+"
RIBBON CABLE CONNECTIONS
MICROPHONE CONNECTIONS
RESISTOR
CAPACITOR
POTENTIOMETER
WIRE

MICROPHONE/AMPLIFIER CIRCUIT BOARD (TOP, COMPONENT SIDE)

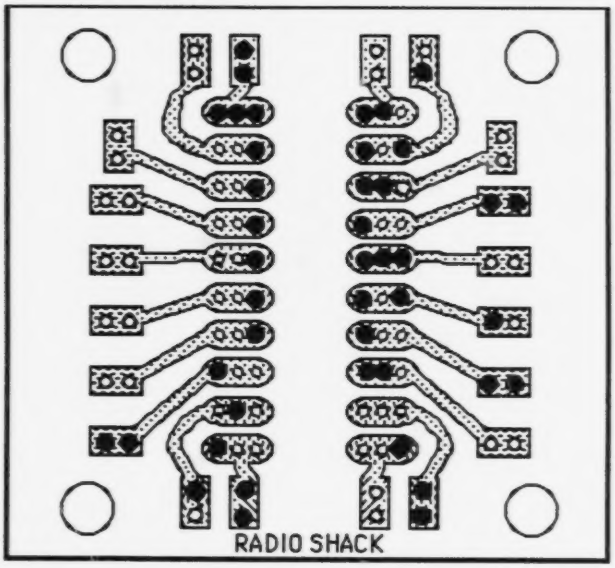
---

[6] The capacitors recommended for this project are not polarized; that is, there is no "+" side to the capacitors.

☐ 5. **Connect the electret microphone.** Notice it has three leads: "+" (a separate wire), "output" (inside of cable), and "ground" (outer shield of cable). You may want to extend the leads on the microphone with extra wire so you can move the microphone around without the circuit board. Connect the three microphone leads at the locations marked "+," "OUTPUT" and "GROUND" on the diagram above.

☐ 6. **Connect the Apple game-port cable to the circuit board.** Three leads on the cable are needed: +5V, GND and PB2. Refer to *Appendix A* for information on identifying these three leads. Strip one or two millimeters of insulation from each wire. Solder the three leads to the locations labeled +5V, GND and PB2 in the diagram above.

☐ 7. **Double-check your circuit.** The diagram below shows the reverse side of the circuit board. It can be used as a quick double check of your circuit. The pattern of solder-filled holes on your circuit board should match this diagram.



MICROPHONE/AMPLIFIER CIRCUIT BOARD (BACK, COPPER-PLATED SIDE)

☐ 8. **Place the IC into the socket.** Carefully insert the LM3900 IC into the socket. Be sure to place its pin #1 in the proper spot and make sure that all 14 pins are inserted properly.

☐ 9. **Test the circuit.** Turn off the computer. Connect the cable to the Apple game port. Place the *Project Programs* disk in the disk drive, and turn on the computer. Place your finger on the top of the LM3900 chip. If it is hot, or if you smell something burning, turn off the computer. You have a wiring error. Refer to the "Troubleshooting" section below. If there are no overheating problems, run the program FREQUENCY. This program should display the frequency of the sound being detected by the microphone. Try whistling into the microphone. Adjust the 50-kΩ potentiometer until the system responds best. If you whistle a steady pitch, a nearly constant frequency should be displayed.

☐ 10. **Check the calibration of your frequency meter.** If possible, check the accuracy of the computerized frequency meter by using some sounds of known frequency. Some sound sources are: tuning forks that are labeled by frequency, audio oscillators with accurate frequency control, or tuned musical instruments.[7] How accurate is the system? Is the accuracy the same for the entire audio-frequency range? How much does the displayed frequency vary when a note of constant frequency is being played?

☐ 11. **Use the system with a musical instrument.** For testing, use a musical instrument that can produce a musical scale, such as a recorder, fife, flute, violin, or even your own whistle (if you have some musical ability). Stringed instruments that are plucked, such as guitars and autoharps, do not work very well since their sound dies out fairly quickly. As mentioned in the "Background Information" section, the system will not work very well with most singing voices or with other sound sources that produce a lot of overtones.

Practice with the instrument until you can produce steady frequency readings for a particular note. Now play a standard diatonic musical scale (do, re, mi, fa, so, la, ti, do'), and write down the frequencies of each of the notes. Calculate the ratio of each of the notes to the first note (do) of the scale. Complete the chart below. Compare the ratios of frequencies that you measure with the theoretical ratios for a diatonic scale.

Note	Theoretical Ratio		Your Measured Values	
	(fraction)	(decimal)	Frequency	Ratio
do	1	1.000	________	________
re	1 1/8	1.125	________	________
mi	1 1/4	1.250	________	________
fa	1 1/3	1.333	________	________
so	1 1/2	1.500	________	________
la	1 2/3	1.667	________	________
ti	1 7/8	1.875	________	________
do'	2	2.000	________	________

How close does the instrument you used approximate the theoretical frequencies of the diatonic scale?

TROUBLESHOOTING

If your microphone/amplifier is not working, follow this procedure:

◊ Double-check your wiring of the circuit board. Carefully compare both the front and back of your circuit board with the diagrams in the "Step-by-Step Instructions."

◊ Make sure you are using the correct lead wires on the cable to the game port. You should be using only +5V, PB2 and GND. Refer to *Appendix A* and use an ohmmeter to make sure you are using the correct wires.

◊ Check for accidental "solder bridges" between pins or solder pads on the circuit board.

◊ If no obvious mistakes are spotted, check the circuit with a voltmeter. Connect the "-" lead of the voltmeter to the GND lead from the game port and do the following:
- Verify the +5 volts at pin #14 and 0 volts at pin #7 of the LM3900.
- Check the voltage at pin #5 of the LM3900. It should be about 2.5 volts.

◊ If possible, use an oscilloscope to see if there is any signal at the three test points listed below:
- Pin #6 of the LM3900: There should be a very weak signal if the microphone is working.
- Pin #5 of the LM3900: There should be about 2.5 volts steady here if there is no sound. An amplified signal reaching 5 volts peak-to-peak centered at 2.5 volts should be present when a loud whistle is fed to the microphone.

---

[7] If you have access to another frequency meter, you can check the calibration of this system by comparing frequency readings with it.

- Pin #4 of the LM3900: This should be the "squared-up" version of the signal. It should be a 0 to 5 volt square wave, which changes frequency as the pitch of the input sound changes.

PROJECT EXTENSIONS

Program Improvements

Modify the FREQUENCY program by adding new features and options. Some suggestions are listed below:

◊ Change the program to save frequency readings in memory so they can be examined or printed later.
◊ Add a large character display for the frequency. Create large block letters using low-resolution graphics.
◊ Modify the program so it can calculate and display the ratios discussed in step #11 of the "Step-by-Step Instructions."
◊ Add graphing of frequency versus time to the program.
◊ Modify the FREQUENCY program or write a totally new one that is controlled by sound. For example, you could write a program that takes different branches, depending on the frequency of your whistle.
◊ Write a program to make the Apple act like a sound-sensitive burglar alarm. When it hears a noise, it should take some action. If you do Project 11 (*Optically-Isolated Switches*), you can have the computer turn on some kind of an electrical device, such as an alarm or a light.

Additional Experiments

Investigate the following:

◊ Using a wind instrument, how much can the frequency of a given note be changed by changing the shape of your mouth or the pressure of your breath?
◊ How close to perfect tuning are the musical instruments that you have available?
◊ What is the highest frequency your ears can detect? If you have a variable-frequency oscillator available, connect it to a speaker so that you can produce a tone. Try increasing the frequency (while measuring it with the FREQUENCY program) to determine the range of your hearing. Keep in mind that either the speaker you are using or the electret microphone may stop functioning before your ears do. The electret microphone is supposed to function up to about 15,000 Hz. It may work at somewhat higher frequencies.
◊ What is the difference between the diatonic and the equally-tempered musical scales? Do some reading on this question and then investigate available musical instruments to see if they are tuned to the diatonic or the equally-tempered scale.[8]
◊ Test how accurately you or your friends can whistle specific musical intervals. For example, the easiest interval to whistle is an octave. Whistle a low note and determine its frequency. Now try to whistle one octave higher. This note should have exactly twice the frequency. Try the same test with other musical intervals (perfect fifth, perfect fourth, major third, etc.).
◊ See if you can find someone with "perfect pitch." Some people are able to whistle a note of a particular frequency on request. See if any of the musicians you know have this ability.

---

[8] Many physics texts include a discussion of musical scales, for example, *College Physics* by Sears, Zemansky and Young, Addison-Wesley Publishing Co., Reeding, MA, 1974.

SUPPLEMENTAL INFORMATION

Program Design Notes

The FREQUENCY program on the *Project Programs* disk is a short program that determines the frequency of the square wave being received on the PB2 input line. Here is a line-by-line explanation of the key parts of the program:

```
130 HIMEM: 32512
```

This sets the upper limit of RAM memory that the BASIC program can use. It prevents the BASIC program from overwriting the machine-language program FREQ, which is loaded into memory starting at memory location 33536 ($8300).[9]

```
150 PRINT D$; "BLOAD FREQ, A$8300"
```

Loads the machine-language FREQ program into memory from the disk.

```
155 POKE 252,2 : POKE 253,0 : POKE 206,151 : POKE 238,1 : POKE 235,0
```

These commands set several memory locations that control the FREQ program. They set the length of time for pulses to be counted and other parameters.

```
200 CALL 33536
```

This line transfers control from BASIC to the machine-language FREQ program, which begins at memory address 33536. The FREQ program counts the pulses coming into the PB2 line for 0.501563 seconds. The result is stored in memory locations 250 and 251 ($FA and $FB in hexadecimal notation). Each number in memory location 251 represents 256 counts.

```
210 C = PEEK (250) + 256 * PEEK (251)
```

After the machine-language routine has finished counting, control returns to BASIC and this statement is used to read the count.

```
220 F = INT (C/0.501563 + .5)
```

The frequency is calculated by taking the counts registered and dividing by the time period over which the counting took place. Using the FREQ program, that time interval is always 0.501563 seconds. The result is then rounded to the nearest integer.

```
230 PRINT C; TAB(13); F
```

Displays both the number of counts registered and the calculated frequency.

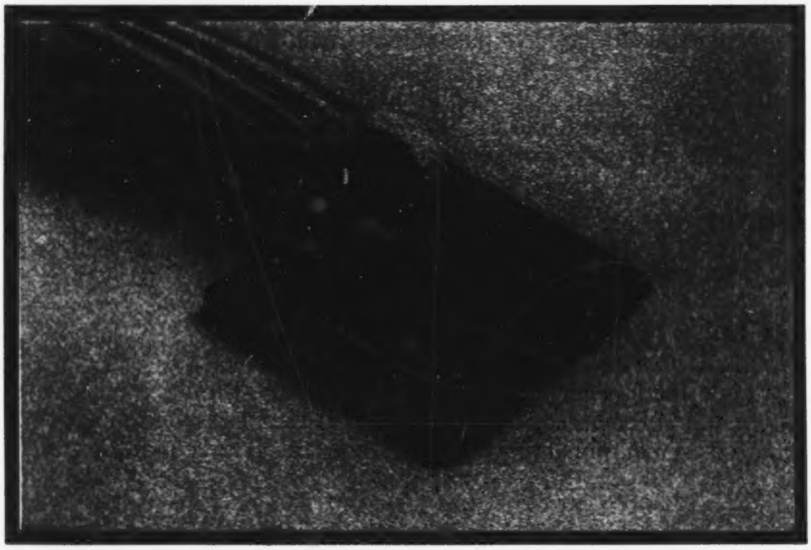
```
270 IF PEEK(49152) < 128 THEN 200
```

Repeats the whole process unless a key is pressed. (If any key is pressed, the value stored in memory address 49152 will increase to greater than 127.)

---

[9] HIMEM is set at $7F00 instead of $8300 (33536) so that the program FREQUENCY will run without modification if it is converted to ProDOS. This allows room for a 1k-byte memory buffer that is needed in ProDOS.

PROJECT 4:
HUMIDITY METER



This project consists of building a circuit to measure the relative humidity of the air. The heart of the circuit is a device specifically designed to change capacitance as the moisture content of the air changes. The circuit produces an output signal that changes from a lower to a higher voltage at regular intervals. The frequency of these transitions varies with the relative humidity. Programs are provided on the *Project Programs* disk that allow the computer to measure this frequency and to calculate and display the relative humidity. The calibration of the computerized humidity meter can be checked by comparing it with other humidity measurements made in other ways.

PROJECT REQUIREMENTS

Background Knowledge

- Electronics: Experience with soldering components on a circuit board is recommended.
- Programming: Some experience with BASIC programming is recommended. Calibrating the project may require changing one line of a BASIC program and then saving the changed version of the program. Some of the extensions require experience with BASIC programming.

Parts

- ☐ Relative Humidity Sensor, Mepco/Electra part number 232269190001 (This part is sold by Mepco/Electra through electronic-parts distributors. Call Mepco/Electra at (817) 325-7871 for the name of the distributor nearest you.)
- ☐ cable to the Apple game port (see *Appendix A*)
- ☐ 555 timer integrated circuit (RS #276-1723)
- ☐ 8-pin IC socket (RS #276-1995)
- ☐ circuit board (Any small circuit board designed for use with ICs will work. One possibility is one-half of a RS #276-159.)
- ☐ 0.01-µF metal-film capacitor, 10 volts or more (RS #272-1051)

☐ 100-kΩ resistor, 1/4 watt, (Use metal film [1%] resistors for this project, if possible. Both resistors needed for this project are included in the Radio Shack metal-film resistor assortment [RS #271-309]. Carbon-composition [5%] resistors can be used, but the calibration of the humidity sensor may not be as good.)

☐ 1-MΩ resistor, 1/4 watt

☐ miscellaneous wires for connections on the circuit board

Note: A "Humidity Meter Parts Kit" containing all of the components needed for this project, including a 16-conductor ribbon cable to the Apple game port, is available from Vernier Software. This kit sells for $23.00. An order form is provided at the back of this book.

Tools and Supplies

Solder, soldering iron, wire cutters and wire strippers.

BACKGROUND INFORMATION

Relative humidity is defined as the amount of moisture present in the air as compared to the amount that it could contain at that temperature and pressure. It is expressed as a percent; for example, if the air contains one-half as much moisture as it could under the given conditions, the relative humidity is reported as 50%.

The relative humidity is an important factor in the weather. It affects the rate of evaporation of water. Water evaporates fastest when the relative humidity is low. This can affect our personal comfort, because our body uses evaporation as a way to keep cool. On hot summer days we feel especially uncomfortable if the relative humidity is high because perspiration does not evaporate very quickly from our skin. On the other hand, low relative humidity inside heated homes in the winter can be a problem. We feel colder in a room with low relative humidity as compared to a room at the same temperature with higher humidity. Dry air is also a frequent source of respiratory problems caused by the drying out of the air passages by too much evaporation.[1]

The relative humidity also influences the amount of static electricity we notice around us. Static electrical charges are separated when dissimilar objects rub together. These charges never get a chance to build up when the humidity is high because the air itself slightly conducts electricity and allows charges to "leak off" almost immediately. When there is less moisture in the air, the charges can increase until a visible (and even audible) spark can be created at discharge. This spark is a short pulse of very high-voltage electricity. It can sometimes destroy an electronic circuit.[2]

An instrument designed to measure relative humidity is called a *hygrometer*. There are several popular types:

- Wet-bulb/dry-bulb hygrometers (also called *psychrometers*): These devices are essentially two thermometers, with the bulb of one thermometer kept wet by wrapping it in a damp cloth. Air is moved across the two thermometer bulbs and the two temperatures are measured. Evaporation causes the wet-bulb reading to be lower than the dry-bulb reading. Tables are available that allow you to determine the relative humidity from the two temperature readings. We will use the wet-bulb/dry-bulb method of measuring the relative humidity as a way of checking the calibration of our computerized humidity meter.

---

[1] For an interesting discussion of the affect of the relative humidity on people and animals, see *The Weather Almanac*, 4th Edition, edited by James A Ruffner and Frank E. Blair, Gale Research Co., Detroit, 1984.

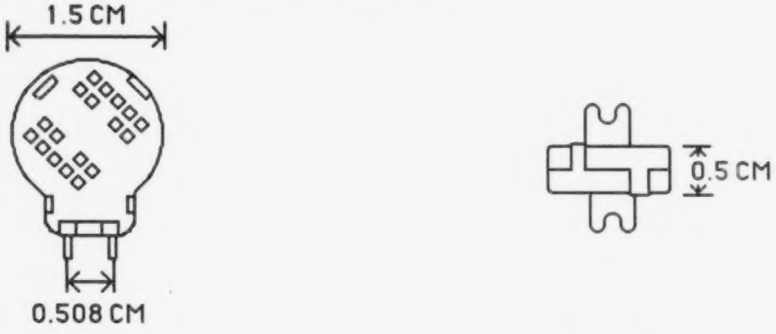
[2] Devices containing integrated circuits that are made using CMOS fabrication techniques are especially sensitive to static-electrical discharges.

- Dew-point hygrometers: If a sample of air is cooled, eventually the moisture in the air will condense, forming droplets of dew. The temperature at which the dew forms is called the *dew point*. If the dew point is measured, tables can be used to determine the relative humidity from the dew point and the original air temperature. One of the project extensions is a simple experiment to measure the dew point. This can be used as another way of checking the calibration of the computerized humidity meter.

- Devices that change physical dimensions due to the absorption of moisture: Many objects change size as the relative humidity changes. One example is a human hair. A hair mounted under tension will expand as the relative humidity increases. This change in length can cause a dial on a humidity meter to change. Some hygrometers used in inexpensive "weather stations" use this method. In general, this type of hygrometer is not very accurate.

- Devices that change electrical characteristics with relative humidity: Electronic components are available that change resistance or capacitance with relative humidity. One advantage of these devices is that they produce an electrical signal that can be fed into a computer for data collection and analysis.

The system used in this project falls into the last category listed above. The specific sensor used is called a *Capacitive Relative Humidity Sensor.* It is manufactured by Philips Electronics. It consists of a plastic case containing a stretched, nonconducting membrane. This membrane is coated on both sides with a very thin coating of gold. The diagrams below show the physical appearance and dimensions of the Capacitive Relative Humidity Sensor.



SIDE VIEW TOP VIEW

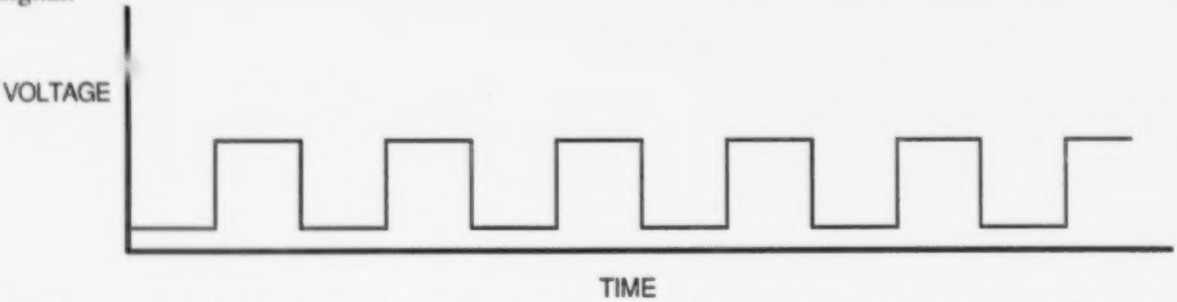
CAPACITIVE RELATIVE HUMIDITY SENSOR

The two gold coatings separated by the insulating membrane form a parallel plate capacitor. The capacitance of such a capacitor depends on the size of the plates (gold coatings in this case), the separation of the two plates, and the *dielectric constant* of the insulating material between the plates. The dielectric constant depends on the electrical characteristics of the material between the plates. The higher this dielectric constant, the larger the capacitance. The dielectric constant of the membrane in the Capacitive Relative Humidity Sensor changes as the relative humidity changes. This causes the capacitance to change in a consistent way with relative humidity. Empirical equations have been developed by the manufacturer that allow you to calculate the relative humidity if you measure the capacitance of the sensor.[3] This equation is listed and discussed in the "Supplemental Information" section below. If the computer can somehow sense the capacitance of the humidity sensor, the computer can then use the empirical equation to calculate the relative humidity. The important thing is that if the capacitance of the sensor is known, there is a way to calculate relative humidity.

---

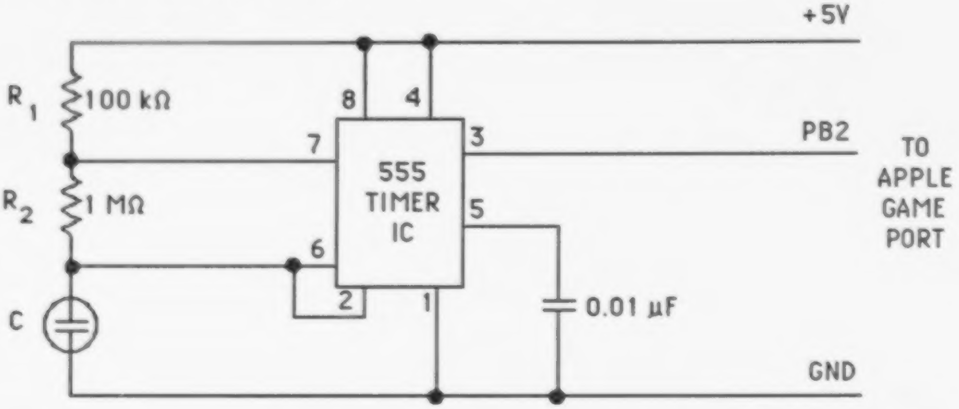
[3] An *empirical* equation is a mathematical relationship that has been determined by studying experimental results. This contrasts with *derived* equations, which are developed by studying the scientific principles involved.

We will use an indirect method to allow the computer to determine the capacitance. The Capacitive Relative Humidity Sensor will be placed in an *astable-multivibrator* circuit. An astable-multivibrator circuit produces an output voltage that varies from high to low at regular intervals. The output signal is referred to as a ***square wave***, because of the appearance of the voltage versus time graph of the signal.



The *frequency* of such a wave is measured by the number of high-voltage pulses produced per second.

The astable-multivibrator circuit uses an integrated circuit called a 555 timer. These are very common, inexpensive ICs used in many electronic circuits. A schematic diagram of the complete astable-multivibrator circuit is shown below.



The output of this astable multivibrator is connected to the PB2 input line on the game port. The frequency of the output is controlled by the capacitance of the device labeled C in the diagram. There is an equation that allows you to calculate the capacitance of device C if you know the frequency of the square wave that is being produced.[4] We will use the Capacitive Relative Humidity Sensor as device C. Since the capacitance of the sensor is affected by the relative humidity, this in turn controls the frequency of the astable multivibrator's output. The computer can determine the frequency of the astable multivibrator's output by counting the pulses that are received in a particular time interval. The machine-language program FREQ is used to do the actual frequency determination. This is the same program that is used in Project 3 (*Microphone/Amplifier*).

Two fairly complicated mathematical equations are used in the process of converting a frequency reading from the astable multivibrator to a capacitance, and then to a relative-humidity reading. These calculations would be very time-consuming if you had to do them by hand or even with a calculator. However, they can be done very efficiently and quickly with a computer. The program HUMIDITY on the *Project Programs* disk takes the frequency reading and does the calculations needed to display the capacitance and the relative humidity.

---

[4] This equation is also listed and explained in the "Supplemental Information" section below.

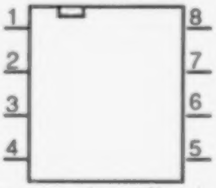
The humidity readings made with this system are typically accurate to within 5%. While this may not seem very exact, it is better than most humidity meters found on home weather stations. This humidity meter can be used in the range 10% to 90% relative humidity. The calibration of the system can be checked by comparison with other relative-humidity measurements.

The HUMIDITY program has been kept simple, so that there is a lot of room for improvements. These will make good project extensions.

STEP-BY-STEP INSTRUCTIONS

☐ 1. **Prepare the cable to the Apple game port.** See *Appendix A* for information on the type of cable you should use. Only three leads will be used on the cable: +5V, PB2 and GND. Strip the insulation from the ends of these three wires.

☐ 2. **Solder the 8-pin socket to the center of the circuit board.** Place the socket on the side of the circuit board without the copper plating. Do not place the IC in the socket. Solder all 8 of the pins to the copper plating on the circuit board. One corner of the socket should have some kind of mark indicating that it is for pin #1 of the IC. Make note of which hole in the socket this is. The pins of an IC socket are referred to by number, and it is essential that you identify the pin holes correctly. The numbering is in a counterclockwise direction, starting at pin #1 as you view the socket from the top side (with the pins pointing away from you).



Note: This is one project that does not work well when assembled on a solderless prototyping board. The reason for this is that the prototyping board itself acts as an extra capacitor that is added to the circuit. This changes the output of the astable multivibrator.

☐ 3. **Add the two resistors to the circuit.** Solder the 1-MΩ resistor so that it connects be - tween pins #6 and #7 of the IC socket. Solder the 100-kΩ resistor so that it connects between pins #7 and #8 of the socket. The diagram below in step 6 shows the resistors in place on the circuit board.

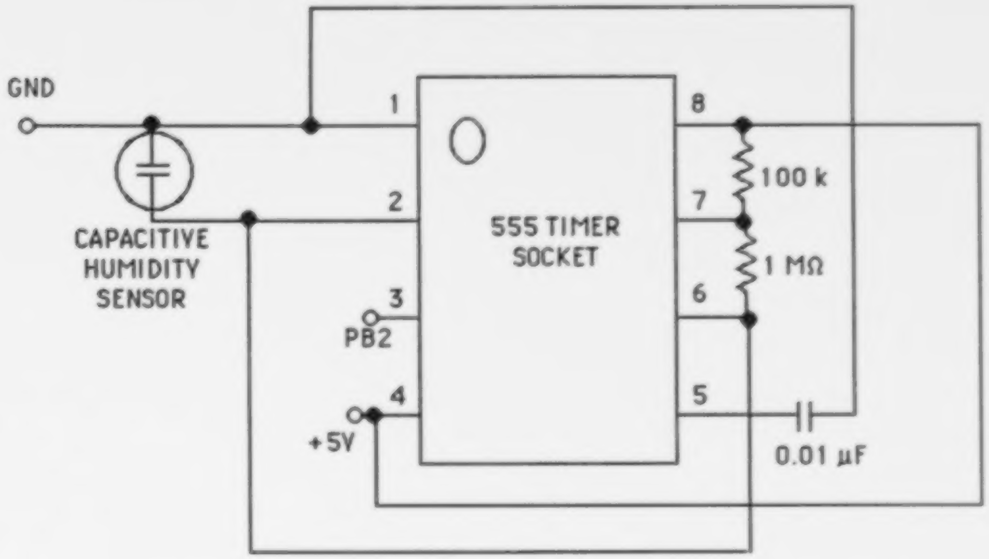
☐ 4. **Solder the 0.01-μF capacitor to the circuit.** The capacitor should connect between pins #1 and #5 of the IC socket. Bend its leads so that they go around one side of the socket.

☐ 5. **Carefully solder the humidity sensor to the circuit board.** The sensor should be soldered between pins #1 and #2 of the IC socket. It may be oriented either way (there is no "+" side). You may have to experiment a little to find two holes with the right spacing for the leads on the sensor. If necessary, bend the leads slightly.[5] **Be careful not to overheat the sensor when soldering.** Use a low-power soldering iron and solder fairly quickly to minimize the heat transfer to the sensor body.

☐ 6. **Add two jumper wires to the circuit.** Add a short section of wire to connects pins #2 and #6 and a second wire to connect pins #4 and #8. The circuit should look like this at this point:

---

[5] If you want to use the sensor at a distance from the computer (for example, to measure outdoor relative humidity), use a long cable between the computer and the circuit board. Do not use a long wire between the IC socket and the sensor. The capacitance of the lead wires will influence the readings.



WORKING DIAGRAM OF THE HUMIDITY METER (TOP VIEW)

☐ 7. **Connect the leads from the game port cable**. The three leads on the game port cable should be attached at the points labeled +5V, GND and PB2 on the diagram above.

☐ 8. **Plug in the IC.** Carefully place the 555 timer in the socket. Make sure to orient pin #1 at the proper corner of the socket.

☐ 9. **Test the humidity-meter circuit.** Turn off the computer. Connect the humidity-meter circuit to the game port. Turn on the computer and run the program HUMIDITY from the *Project Programs* disk. The program should display both the capacitance of the sensor and the relative humidity. Try breathing on the sensor. This should cause a gradual increase in the relative-humidity reading. Notice that the sensor takes a few minutes to respond to changes in humidity.[6] If the capacitance displayed is not in the range 100 pF to 150 pF, consult the "Troubleshooting" section below.

☐ 10. **Check the calibration of the meter.** If you have an accurate, calibrated hygrometer available, simply compare the relative-humidity reading from the computer and the reading from the hygrometer. If you do not have a hygrometer, you can improvise a wet-bulb/dry-bulb hygrometer. All you need is a good quality thermometer (preferably Celsius) and the *Handbook of Chemistry and Physics*[7] or other reference book that includes a wet-bulb/dry-bulb relative-humidity table. First take the temperature of the air in the room using the dry thermometer. Next, moisten a piece of muslin or cotton gauze with clean, room-temperature water and place the cloth around the bulb of the thermometer. Vigorously wave a sheet of paper as a fan near the thermometer for several minutes. Now take the wet-bulb reading of the thermometer. Calculate the difference in the two temperature readings. Most wet-bulb/dry-bulb relative-humidity tables use the difference in temperatures and the dry-bulb temperature. Determine this relative-humidity value from the table and compare it with the reading from the computer. If the computerized humidity meter reading and the relative humidity measured with the thermometer differ by more than 5 or 10%, refer to the "Troubleshooting" section below.

---

[6] The maximum response time for the sensor is listed in Mepco/Electra Data Sheet ME RHS-583 as 5 minutes for relative humidity greater than 43%, and 3 minutes when the humidity is lower .

[7] *Handbook of Chemistry and Physics*, published by the Chemical Rubber Company, is a standard reference book found in many public and school libraries.

Investigate the reproducibility of the system: check the reading, breath on the sensor to increase the humidity, and then wait to see if the reading eventually returns to the original value. This may take several minutes.

☐ 11. **Monitor the relative humidity over a period of several days.** Set up a data sheet similar to the one below.

DATE	TIME	INDOOR RELATIVE HUMIDITY (FROM COMPUTER)	INDOOR RELATIVE HUMIDITY (FROM DRY-BULB/ WET-BULB READINGS)	OUTDOOR RELATIVE HUMIDITY (FROM WEATHER REPORTS)	COMMENTS

Find a local radio or television station that regularly reports the relative humidity as part of the weather report. Begin collecting data on the outdoor relative humidity (from the weather reports) and the indoor readings taken with your computerized humidity meter and from wet-bulb/dry-bulb readings (as in step #10). Make note of factors that might affect the indoor relative humidity in the "comments" section of the table. Some examples might be:

furnace is on most of the time during a cold spell
great weather - windows open all day
brother has big date - takes 5 showers to get ready

After several days of taking data, try to answer some of the questions below:

- How accurate is your computerized humidity meter? Does it seem to be accurate over the entire relative-humidity range that you tested?
- Is the relative humidity indoors consistently higher or lower than the relative humidity outside?
- How does the relative humidity change during the course of a "normal" day? Why is this?
- Are you able to sense the relative humidity from the way the air "feels"?
- Can relative-humidity readings be used to predict the coming weather?

TROUBLESHOOTING

If the capacitance readings are unreasonable (not between 100 and 150 pF):

A capacitance reading outside the range of 100 to 150 pF indicates a major problem with the circuit. Check the following likely troublespots:

◊ Are your cable connections correct? Refer to *Appendix A* and check your connections with an ohmmeter.

◊ Is the 555 timer IC inserted in the socket correctly? Make sure pin #1 is in the right hole. Are all 8 pins properly seated in the socket?

◊ Are the two resistors and the capacitor in the right locations? If you have not located the problem, double-check all of the connections. If this still does not work, try replacing the Capacitive Relative Humidity Sensor with a fixed-value 100-pF capacitor. When unsoldering the sensor, be careful not to overheat it. The circuit should produce an output signal with the 100-pF capacitor in place. The capacitance reading displayed when the HUMIDITY program is run should be close to 100 pF. If this test fails, there is something wrong with the circuit. If the circuit works with the capacitor substituted for the humidity sensor, there is probably something wrong with the sensor.

◊ Have you used a prototyping board to assemble this project or have you used long wires between the humidity sensor and the circuit board? If so, the circuit may not work correctly.

If your relative-humidity meter is not calibrated correctly, but the sensor responds:

If the readings displayed by the HUMIDITY program are consistently too high or too low, but the sensor responds when you breathe on it, you need to change the calibration of the sensor. You can do this by making a change in the HUMIDITY program. You need to change the value of C12 that is assigned in line 260 of the program. If you want your computerized humidity sensor to read higher, you need to **decrease** the value of C12. This constant is supposed to be the capacitance of the sensor when the relative humidity is 12%. It is normally 112 pF, and this is the value assumed in the HUMIDITY program. It is possible that the particular sensor you are using has a different value. If you want the readings to be lower, **increase** C12. The simplest way to do this is by a "trial-and-error" method. Change the value by 1 and run the program again. Repeat this with different values until the system is properly calibrated. After you have the calibration adjusted, save the HUMIDITY program on the disk with the new value of C12 so that it will read correctly the next time you use it.

PROJECT EXTENSIONS

Program Improvements

There are many improvements that can be made to the HUMIDITY program. Some suggestions are:

◊ Change the program so that it can collect data in memory and display it on the monitor or print it later.
◊ Allow the relative-humidity data to be saved on a disk.
◊ Have the program keep track of and display the time as it takes relative-humidity readings.
◊ If your computer has a "clock card" to keep track of real time, change the program so that it reads the clock time and stores it along with the relative humidity. This would be very useful for a weather station.
◊ Change the program so that relative-humidity readings at preselected time intervals (for example, every hour) are saved in memory. Again, this would be desirable for a weather station.
◊ Add a mode to the program that graphs the relative humidity versus time.
◊ Add a mode to the program that displays the relative humidity in large block digits so they can be read at a distance.

Circuit Modifications

◊ Mount the humidity-sensor circuit in a protective housing. The Capacitive Relative Humidity Sensor is fairly fragile, so it should be somewhat protected. At the same time, it cannot be sealed off from the air flow. Either mount the sensor inside a box with holes in it or mount it inside a box and leave off the top.

◊ Combine this project with one or more IC Temperature Probes (from Project 6) to make a weather station. The humidity meter uses only the PB2 input line at the game port, so it can be combined with the circuit for the integrated-circuit temperature sensor (which uses the PDL0 line) with no conflict. In fact, several temperature sensors can be used with the humidity meter. Note, however, that you will need to do some changing (or combining) of the programs that read the signals from the sensors. If you are willing to do a lot of work on your own, you may be able to design and connect sensors that measure wind speed, wind direction, or air pressure.

◊ Replace the Capacitive Relative Humidity Sensor with various capacitors and use the circuit as a capacitance meter. The HUMIDITY program displays the capacitance of whatever is connected between pins #1 and #2 of the 555 timer IC. With the program and the circuit as originally wired, this system could only be used to measure a narrow range of capacitance. If the resistors R_1 and R_2 are changed, the range of this capacitance meter can be expanded. The HUMIDITY program will also need to be changed so that it correctly calculates the capacitance. It should use the equation discussed in the "Calculation of Relative Humidity" section. Refer to *The 555 Timer Applications Sourcebook* by Howard M. Berlin, Howard W. Sams & Co. (1976) for more information.

Additional Experiments

◊ Check the calibration of your relative-humidity meter by measuring the *dew point*. The dew point is the temperature to which the air must be cooled before the water vapor in it begins to condense, thereby forming dew. If the dew point is measured, you can determine the relative humidity from reference tables. This gives you another method of checking the accuracy of your computerized humidity meter.

To measure the dew point, you will need a thin-walled container made of shiny metal. A "tin" food can with the label removed will work reasonably well. Fill the can half-full of room temperature water. Gradually cool the water in the container by adding small pieces of ice. Stir the water continuously (but gently) with a thermometer. The stirring is to insure that all of the water and the container are at the same temperature. Watch the outside, lower part of the can for the first signs of condensation. This process must be done slowly—add a piece of ice, stir, look for condensation; add more ice, stir, etc. As soon as a faint mist appears, note the water temperature.[8] Avoid breathing on the can, because your breath contains a lot of extra moisture that can affect the reading.

Remove any remaining ice from the water. Slowly warm the water (and the container) by adding warm water while stirring constantly. Carefully observe the container and note the temperature at which the moisture disappears.

The average of the two temperatures recorded is the best estimate of the dew point.[9] The difference between the air temperature and the dew point should also be calculated. This is called the *depression of the dew point*. Locate a relative-humidity/dew-point table in a reference book, such as the *Handbook of Chemistry and Physics*. This table will allow you to find the relative humidity from the measured values of the dew point and the depression of the dew point.

◊ Calculate the *Temperature-Humidity Index (THI)*. This number scale is used to express the combined temperature-humidity effect of warm weather. The higher the number, the more discomfort people feel. Formulas are available to calculate the THI value from the relative humidity and the temperature.[10] If you combine this project with Project 6 (*IC Temperature Probe*), you could even have the computer calculate and display the THI values.

◊ If you have access to a humidifier or a dehumidifier, use the humidity meter to see how well it works. Place the appliance near the humidity sensor and see how effective it is in changing the relative humidity of the air.

◊ The Capacitive Humidity Sensor can also be built into a "stand-alone" humidity meter. Complete plans for such a meter are included in "Build This Humidity Monitor" by Mark C. Worley, *Radio–Electronics*, February 1986, pp. 61-70.

---

[8] One way to notice the first formation of dew is to look at the reflected image of text in the surface of the container. As soon as dew forms, the letters will get blurry.

[9] For a better estimate, repeat the experiment several times and average all of your results.

[10] Refer to *The Weather Almanac*, 4th Edition, edited by James A Ruffner and Frank E. Blair, Gale Research Co., Detroit, 1984, pp. 127-128.

SUPPLEMENTAL INFORMATION

Calculation of Relative Humidity

The process of determining the relative humidity involves some complicated calculations. Fortunately, these calculations are done automatically by the HUMIDITY program. The mathematics of the process is explained here for those who are interested in seeing how this is done.

The frequency produced by the *astable-multivibrator* circuit used in this project is determined by the capacitance of the device connected between pins #1 and #2 of the timer IC and by the values of the two resistors used in the circuit. The equation showing how the frequency is related to these three variables is:[11]

$$f = 1.443 / ((R_1 + 2R_2) \times C)$$

where: C is the capacitance of the device between pins #1 and #2
R_1 is the resistance of the resistor between pins #7 and #8
R_2 is the resistance of the resistor between pins #6 and #7

In this project we used $R_1 = 1\ \mathrm{M}\Omega$ and $R_2 = 100\ \mathrm{k}\Omega$, so the equation becomes:[12]

$$f = 1.443 / ((1 \times 10^5 + 2 \times 10^6) \times C)$$
$$f = 1.443 / (2.10 \times 10^6 \times C)$$

Solving for C,

$$C = f \times 1.443 / (2.10 \times 10^6)$$

This equation is used by the HUMIDITY program to calculate the capacitance of the humidity sensor from the frequency produced by the astable multivibrator.

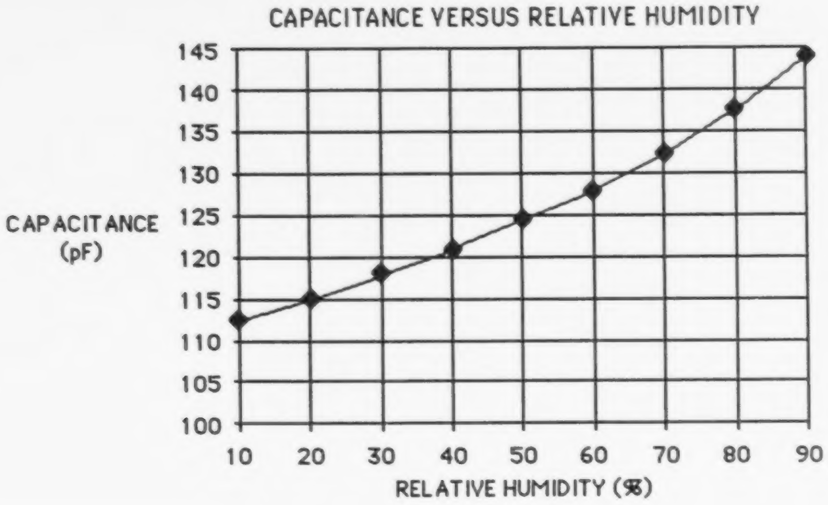
The next step in the process is to calculate the relative humidity from the capacitance. The graph below shows how the capacitance of the humidity sensor varies with relative humidity.[13]

---

[11] This equation is derived in *The 555 Timer Applications Sourcebook* by Howard M. Berlin, Howard W. Sams & Co., Inc., Indianapolis, Indiana (1976).

[12] Metal film (1%) resistors were recommended for this project so that this equation would be as accurate as possible.

[13] This data taken from Mepco/Electra Data Sheet ME RHS-583.



The manufacturer has done some "curve fitting" (probably using a computer) and has shown that the formula below approximates this experimental data:[14]

$$C/C_{12\%} = 0.0985 + 0.34 \times (H/100)^{1.4}$$

where: C = the capacitance of the sensor (in units of pF)
$C_{12\%}$ = the capacitance of the sensor when the relative humidity is 12% (in pF)
H = the relative humidity (in percent)

This equation, solved for H, is used by the HUMIDITY program in calculating the relative humidity. The graph and the equation are for a typical Capacitive Relative Humidity Sensor, and a particular sensor may have slightly different characteristics. The value used for $C_{12\%}$ is normally 112 pF, but it can be adjusted slightly up or down, if necessary, to match an individual sensor's characteristics. This correction factor is discussed in the "Troubleshooting" section above.

The Philips Company N.V., which manufactures the Capacitive Relative Humidity Sensor, lists its accuracy as within 5% over the range 10% to 90% relative humidity. An additional calibration error is introduced when the temperature of the sensor changes. According to Philips Company data sheets, the change in capacitance with temperature is 0.1% per °C. For use indoors, where the temperature does not vary too greatly, this error is not a major problem.

---

[14] This formula is from "Capacitive Humidity Sensor for Consumer Applications," Technical Note 134, Philips Nederland B.V., Eindhoven, The Netherlands, 1983.

PROJECT 5:
RESISTANCE/CAPACITANCE METER

This project demonstrates using the Apple computer as a digital capacitance meter or digital ohmmeter. A meter of this type can be a useful laboratory tool. The project will use a machine-language program to read the status of a circuit connected to the PDL0 line on the game port. A BASIC program called METER is also provided on the *Project Programs* disk to allow both capacitance and resistance to be measured. The project extensions involve improving the METER program and doing experiments with the system.

PROJECT REQUIREMENTS

Background Knowledge

- Electronics: This project requires an understanding of what a capacitor is and what it does. You should also know Ohm's law and simple algebra. The circuit assembly is easy.
- Programming: The core project requires no programming. Some of the project extensions require familiarity with BASIC.

Parts

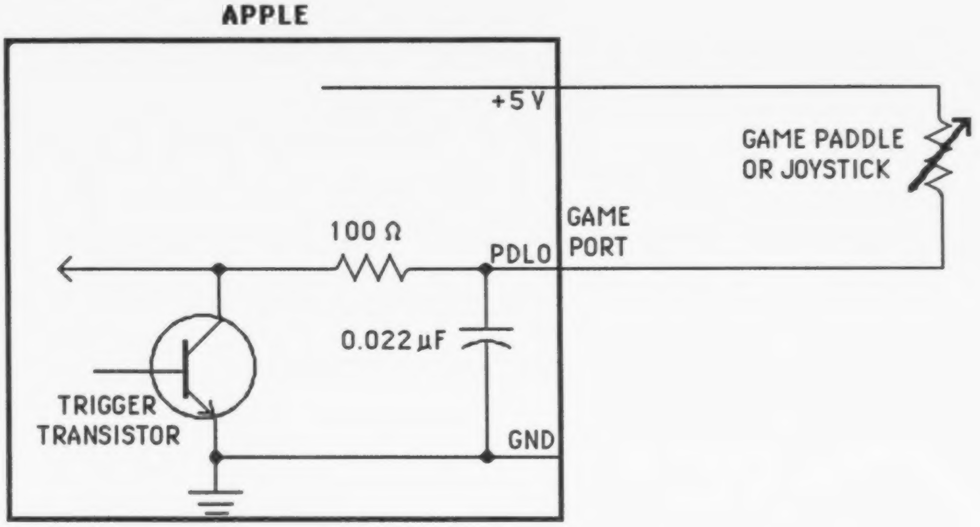
- ☐ 4-position DIP switch (RS #275-1304)
- ☐ cable to Apple game port (see *Appendix A*)
- ☐ dual IC circuit board (Any small circuit board can be used, but the Radio Shack #276-159 is recommended.)
- ☐(2) 8-pin IC sockets (RS #276-1995)
- ☐ 47-kΩ resistor, 1/4 watt, 10%
- ☐ 10-μF capacitor, 10 volts or more
- ☐ short sections of wire for circuit connections
- ☐ miscellaneous resistors for testing (1 kΩ to 30 MΩ, 1/4 watt or more)
- ☐ miscellaneous capacitors for testing (0.01 μF to 10 μF, 6 volt or more)

Tools and Supplies

Ohmmeter, soldering iron, solder, wire cutters and wire strippers.

Background Information

All Apple II computers have PDL input lines on the game-port connector. These lines were originally designed to read the position of a game paddle (hence the label PDL). In normal use, the circuit involving a PDL line looks like this:

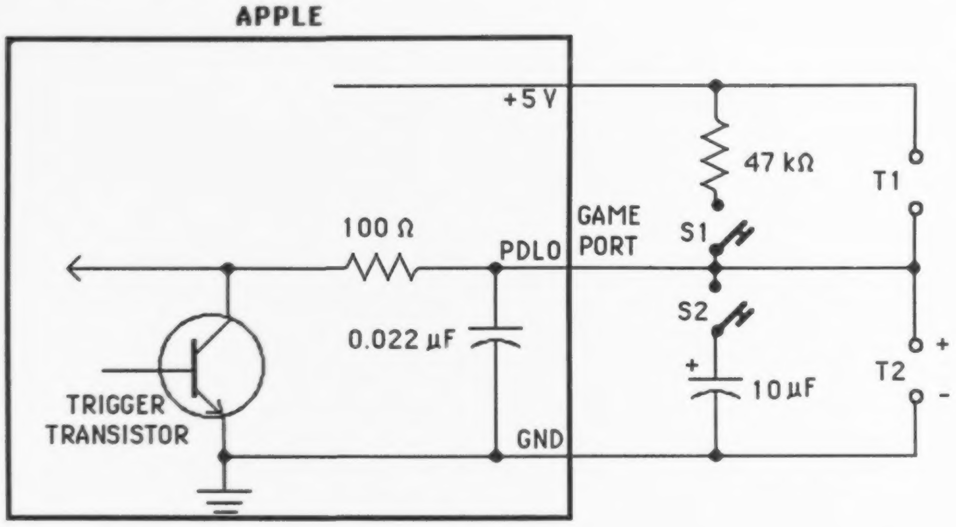


This project uses the PDL input lines in a special way to measure resistance or capacitance. Before you can understand how this project works, you need to know how the PDL inputs are normally used. The "Apple Game Port" section of the *Introduction* explains this in detail. Take a few minutes to read about the PDL input lines in this section before starting this project.

The simplest way to use the Apple to measure resistance would be to replace the game-paddle resistor in the PDL0 circuit with an unknown resistor and use the BASIC PDL(0) command to measure its resistance. The computer would display an integer between 0 and 255 to represent resistance. To calibrate this "ohmmeter" you would have to compare the count readings for various resistors with the actual resistance values (measured with a conventional ohmmeter). Even after this calibration, this system would have some limitations. It could only read one of 256 possible resistance values evenly spaced over the range of 0 to 150 kΩ. Each count number would represent (150,000 Ω) / 256 or about 600 ohms. The results would not be very precise.

In this project we are going to make a few changes in both the hardware and the software to make the meter more precise and to allow measurement of capacitance as well as resistance. First we will replace the built-in Apple paddle-check routine (PDL(0) command) with a machine-language program named PDL.CHECK. This program is on the *Project Programs* disk. It will determine the time it takes the capacitor to become charged by counting loops as the PDL(0) routine did, but it reads from 0 to 40,960 counts. Each loop in this program takes 18 µsec. With this routine, even if it takes a long time (a few tenths of a second) to charge the capacitor, a meaningful count is registered. The counter does not "overflow" when it reaches 256 counts, as the PDL(0) command does. The maximum time is now 40,960 x 18 µsec or about 0.7 sec. The price we pay for this improvement is that checking the PDL line this way takes longer and is no longer as simple as using a PDL(0) command in BASIC. To use the PDL.CHECK program, a BASIC "CALL" statement will be used.

The hardware changes are shown below:



RESISTANCE/CAPACITANCE METER CIRCUIT

The key changes are the addition of an external capacitor and a resistor that can be switched into the circuit. A 47-kΩ resistor can be added to the circuit by closing switch S1. A 10-µF capacitor can be added to the circuit by closing switch S2. There are also two pairs of terminals labeled T1 and T2, where additional components can be plugged in.

With these additions to the circuit, several different measurements can be made using the Apple game port:

- If both switches (S1 and S2) are open, the circuit is the same as the normal PDL input circuit. An unknown resistor can be plugged into the T1 terminals. Its value can be determined by the time it takes the Apple's internal capacitor to charge up. The limitation on this arrangement is that if a small value resistor is used at T1, the time it takes the capacitor to charge may be so short it cannot be measured accurately. Switch S2 helps solve this problem.
- If switch S2 (only) is closed, an extra 10-µF capacitor is added in parallel with the Apple's internal capacitor. This greatly increases the capacitance of the system.[1] An unknown resistor can be plugged into T1 as before, but since more charge is needed, it takes more time for the capacitors to reach the threshold voltage level. Smaller value resistors can now be tested using the T1 terminals.
- If switch S1 (only) is closed, a 47-kΩ fixed resistor is switched into the circuit. An unknown capacitor can then be added to the circuit by plugging it into the T2 terminals. The time it takes this capacitor (along with the Apple's internal capacitor) to charge up with the current through this resistor is measured. This time can be used to determine the capacitance of the unknown capacitor.

Let's take a quick look at the mathematical relationships involved in calculating resistance or capacitance.[2] The computer will actually measure a *count*, which is a time unit measuring the charging time of the capacitor. This time depends on two factors: (1) the current (the rate that charge is flowing) to the capacitor, and (2) the capacitance of the capacitor.

---

[1] When two capacitors are placed in parallel as these two are, the total capacitance is the sum of the two capacitances.

[2] For a more detailed discussion of the mathematical relationships involved in charging capacitors, refer to the "Additional Mathematics Background" section of this project.

The current is controlled by the resistance in the charging circuit. The larger the resistance, the smaller the current, and the longer it will take the capacitor to become charged. **Charging time is proportional to resistance.**

Capacitance is a measure of how much electrical charge a device stores per volt. Capacitance is measured in a unit called a *farad*. A farad is a very large unit, so microfarads (µF) are used more often.[3] The larger the capacitance, the longer it will take for enough charge to accumulate on the capacitor to allow it to reach the threshold voltage level. **Charging time is proportional to capacitance.**

The equation expressing the relationship between capacitance (C), resistance (R) and charging time is:

$$\text{charging time} = k \times C \times R$$

In words, the charging time (in counts) is proportional to the product of R times C. The constant of proportionality (k) depends on the threshold voltage level and the units used. This proportionality constant can be determined by using the computer to measure the charging time with a known resistor and a known capacitor. After this calibration, the system can be used to measure either resistance or capacitance over a wide range.

The equation above will work fine, regardless of the units you use for resistance and capacitance, as long as you are consistent. For example, if you enter all capacitances in µF and all resistances in kΩ, everything will work out well. Do not mix in capacitances or resistances measured in other units. We suggest measuring all resistors in kilohms and all capacitors in microfarads throughout this project.

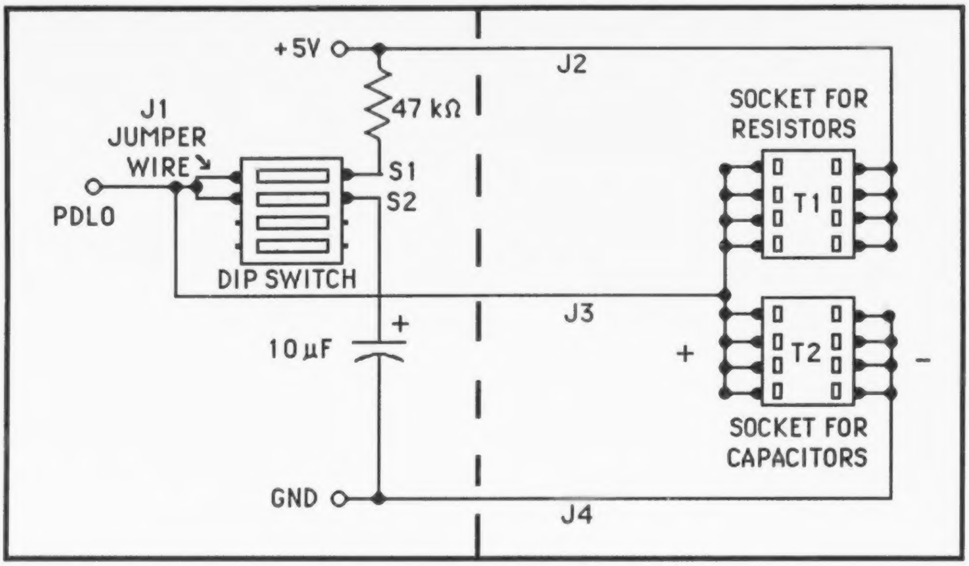
The program METER on the *Project Programs* disk will continuously check and display the charging time in counts. This program is very short and simple. It can be modified (as a project extension) to display resistance and capacitance directly.

STEP-BY-STEP INSTRUCTIONS

A Radio Shack Dual IC Circuit Board (RS #276-159) is recommended for this project. A working diagram of the circuit is shown below. Refer to this diagram during assembly.

---

[3] One microfarad (µF) is equal to 1×10^{-6} farads. Capacitors are sometimes rated in picofarads (pF). One picofarad is equal to 1×10^{-12} farad.



WORKING DIAGRAM OF RESISTANCE/CAPACITANCE METER CIRCUIT (TOP VIEW)

☐ 1. **Solder the DIP switch on the circuit board.** Place the switch unit in the center of the left half of the board on the side that is not copper plated. Switch #1 should be at the top. Solder all 8 pins to the copper-plated side. The DIP switch unit contains four switches. We will use only two of them.

☐ 2. **Add a jumper wire to the switch.** Connect the top two pins on the left side of the DIP switch with a short, non-insulated connecting wire J1, as shown in the working diagram above.

☐ 3. **Prepare the cable to the Apple game port.** Refer to *Appendix A* for information on cables to use and diagrams to help identify the leads. The three leads used for this project are: +5V, PDL0 and GND. Strip a few millimeters of insulation from each of these leads.

☐ 4. **Solder the cable to the circuit board.** Solder the PDL0 lead to the jumper wire on the DIP switch. Solder the +5V line to a point near the top of the left half of the IC board. Solder the GND lead to a point near the bottom of the left half of the board. Other wires will be connected to these points later.

☐ 5. **Determine the exact resistance of the resistor labeled "47 kΩ."** Use a good quality ohmmeter to determine the exact resistance of the 47-kΩ resistor. This value will be used in calibrating the system. Write down the result:

R_{47k} = _____ k Ω

☐ 6. **Solder the 47-kΩ resistor onto the circuit board.** This resistor should be soldered between the right side of switch S1 and the +5V lead.

☐ 7. **Solder the 10-µF capacitor onto the circuit board.** The capacitor goes between the right side of switch S2 and the GND lead. Note that most 10-µF capacitors are *polarized*; that is, they have a "+" and a "-" side. If the capacitor you are using is polarized, make sure the negative side of the capacitor is connected to the GND terminal. This completes the left side of the circuit.

☐ 8. **Solder the two sockets onto the right side of the circuit board.** Place the two sockets on the board, one above the other. Use a bare (not insulated) piece of wire to connect together all 8 of the terminals on the left side of both sockets. The easiest way to do this is to lay a straight, bare wire along all 8 pins and make solder connections between it and each pin. Use a shorter, uninsulated section of wire to connect together the four terminals on the right side of the top socket. Use another short section of uninsulated wire to connect together the four terminals on the right side of the lower socket. Solder the sockets and the connector wires into place. Label these two sockets T1 and T2. Also label the two sides of the T2 socket "+" and "-" as shown in the working diagram.

☐ 9. **Connect the right and left halves of the dual circuit board.** Add three jumper wires (J2, J3 and J4), connecting the left and right sides of the circuit as shown in the working diagram at the beginning of these instructions.

☐ 10. **Connect the circuit to the computer.** With the computer off, plug the circuit into the Apple game port. Turn on the computer. If the computer will not start up normally, turn it off and double-check your circuit. You must have an incorrect connection.

☐ 11. **Run the METER program.** Place the *Project Programs* disk in the disk drive and run the program named METER. This is a simple program that loads the PDL.CHECK routine, reads and displays the count on the screen.

☐ 12. **Determine the proportionality constant k.** Set switch S1 on and switch S2 off, so the 47-kΩ resistor is in the circuit. Read the counts. Assume that the Apple's internal capacitor is exactly 0.022 μF, as it is supposed to be. Since you know the exact resistance of the "47-kΩ" resistor (from step #5 above), you can now calculate the value of k in the equation:

$$\text{counts} = k \times R_{47k} \times C$$

where: $C = 0.022\ \mu F$

R_{47k} = the exact value of your 47-kΩ resistor

Substitute the value of R (in units of KΩ) and calculate k. Write this number down; you will use it later.

$$k = \text{counts} / (0.022\mu F \times R_{47k}) = \_\_\_\_\_$$

☐ 13. **Determine the exact capacitance of your system, including the 10-μF capacitor**. Set both switches S1 and S2 to the "on" position, so that both the 47-kΩ resistor and the 10-μF capacitor are in the circuit at the same time. In this way, we can determine the actual capacitance of the system when the capacitor labeled "10 μF" is used. Read the counts and calculate the capacitance again using the formula: counts = k x R x C. Use the exact value of the "47-kΩ" resistor as R and the value of k determined in the previous step. Write this value down because it will be used often. This is the total capacitance of the system including the "10-μF" capacitor and the Apple's internal capacitor.

$$C_T = \text{counts} / (k \times R_{47k}) = \_\_\_\_\_\ \mu F$$

☐ 14. **Test the system by measuring the capacitance of some labeled capacitors.** Set switch S1 "on" and S2 "off." The capacitance meter should now be ready to use. Place the capacitor to be measured across the T2 terminals with one lead on the left side of the socket and one lead on the right side. It is a good idea to discharge the capacitor before connecting it to the circuit (touch its two leads together). If it is a polarized capacitor, be sure the negative side is connected to the right side of the socket. Capacitors checked should be rated at 10 volts or more (most are). **Do not use capacitors larger than 10 μF.** The capacitance can be calculated from the equation:

$$\text{counts} = k \times R \times C_{total} = k \times R_{47k} \times (C_x + 0.022\ \mu F)$$

The capacitance of the unknown resistor (C_x) is then:

$$C_x = \text{counts} / (k \times R_{47k}) - 0.022\ \mu F$$

Read the counts from the screen. Substitute R_{47k} (the exact resistance of the "47-kΩ" resistor) and k (the constant from step #12 above) and calculate C_x, the capacitance of the unknown capacitor.

Calculate C_x for several different capacitors. Compare your measured values with the values labeled on the capacitors. Make a table of labeled versus measured values. Capacitors are labeled with several different code schemes. Consult an electronics reference book, if necessary.[4]

You can effectively measure capacitances from about 0.01 to 10 µF with this system. Obviously, all of this hand calculation of capacitance using the same equation over and over is a job better done by the computer. That is one of the suggested project extensions.

☐ 15. **Measure the value of some resistors.** For use as a digital ohmmeter, this system has two ranges:

High-Resistance Range: To measure large resistances (30 kΩ to 30 MΩ), set the switches with S1 "off" and S2 "off." Connect the resistor across terminal T1 with one lead on the left side of the socket and one lead on the right side. Read the count and calculate the resistance from:

$$\text{counts} = k \times R \times C$$

This time C is known. It is 0.022 µF, the capacitance of the Apple's internal capacitor, therefore,

$$\text{counts} = k \times R \times 0.022\ \mu F$$

$$R = \text{counts} / (k \times 0.022\ \mu F)$$

Low-Resistance Range: To use the system to measure lower-value resistors (1 kΩ to 30 kΩ), set the switches with S1 "off" and S2 "on." Connect the resistor across the T1 terminals. Calculate the resistance as above, except this time C is the total capacitance of the system with the "10 µF" capacitor included. This is the value measured in step #13 above.

$$R = \text{counts} / (k \times C_T)$$

where: C_T = total capacitance of the system including the 10-µF capacitor

Test the circuit with a number of resistors. Make a table of labeled values and measured values. Resistors are often labeled using the color code shown in *Appendix B*. Do all of the resistors you tested fall within their labeled tolerance limits?

Test the accuracy of the computerized resistance meter by comparing its resistance values with a digital ohmmeter. At what resistances is the system most accurate? What is the maximum error?

Here is a summary chart showing how the two switches should be set for various measurements.

---

[4] One good source is *The Radio Amateur's Handbook* published by the American Radio Relay League.

MEASUREMENT	S1	S2	EQUATION USED
CAPACITANCE (0 - 10µF)	ON	OFF	$C_x = \text{counts} / (k \times R_{47k}) - 0.022\ \mu F$
RESISTANCE (> 30 kΩ)	OFF	OFF	$R = \text{counts} / (k \times 0.022\ \mu F)$
RESISTANCE (< 30 kΩ)	OFF	ON	$R = \text{counts} / (k \times C_T)$ where: C_T = total capacitance from step #13)

TROUBLESHOOTING

If the system does not function as you expect, check the following:

- Have you used the leads from the Apple game port correctly?
- Are your switches working properly? Check for continuity when the switch is "closed" by using an ohmmeter.
- Is your 10-µF capacitor oriented properly (if polarized)?
- Are your resistors and capacitors really of the values you think they are? Check them with a digital meter or try some different components.
- If the count reading is -24576, then either the resistance or the capacitance is too high. Something must not be connected properly.
- If the count reading is very low, then there must be a short between the leads.

PROJECT EXTENSIONS

Program Improvements

The METER program on the *Project Programs* disk is very primitive. Here are some possible improvements and additions you might want to try:

◊ Modify the program to do the calculations of capacitance and resistance. You may want to have three different "modes of operation" that the operator selects, depending on what is to be measured (capacitance, low resistance or high resistance). Build into the program your values of k and the fixed resistors and capacitors. When measuring capacitance, the program should probably subtract the capacitance of the Apple's internal capacitor (0.022 µF), so the result reported is the capacitance of just the unknown capacitor. Have the program round the values displayed to the proper number of significant digits.

◊ Have the program display either resistance or capacitance values in large block digits, so they can be read from a distance. Use Apple low-resolution graphics for the letters. Perhaps you can include the units.

◊ Modify the program so that it draws the resistor's color-code bands, just as they are on the resistor. Use a color video monitor and low-resolution color graphics.

Circuit Modifications

◊ If you want to turn the circuit into a practical tool, mount the circuit in a box. You may want to use something other than 8-pin IC sockets as the terminals for unknown components. Replace the two DIP switches with a single rotary switch that has the following positions:

CAPACITANCE
RESISTANCE (LOW)
RESISTANCE (HIGH)

◊ Separate terminals T1 and T2 from the rest of the circuit so you can run tests on the temperature coefficients of the components plugged into these terminals. Break the two halves of the dual IC circuit board apart and run long (one meter) connecting wires (J2, J3 and J4) between the two halves of the board. This allows you to do tests of the temperature response of resistors and capacitors plugged into the terminals. Heat the component (for example with a hair dryer) and look for changes in the resistance or capacitance. If you do not separate the two halves of the circuit and try to do temperature testing, you may heat the built-in 47-kΩ resistor and 10-μF capacitor. Changes in their values may artificially affect the results.

Additional Experiments

The following experiments can be conducted with the resistance/capacitance meter:

◊ Investigate the effect of combining two resistors or two capacitors in series and in parallel

Measure the resistance of two different resistors. Combine them in series and measure their resistance. Now combine them in parallel and measure their resistance. Repeat this experiment with several other pairs of resistors. If you do not already know the theoretical rule for how resistance "adds up" in these two cases, look it up and report on how well your data agree with the theoretical results. How do your measurements compare with theory?

Repeat this experiment using capacitors instead of resistors. How do your experimental results compare with theory?

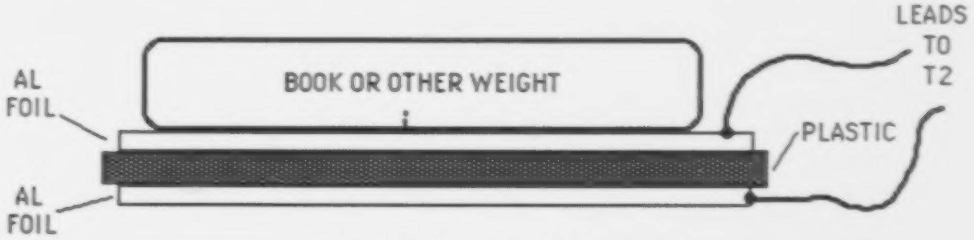
◊ Measure the capacitance of homemade capacitors

A capacitor is essentially two metal plates separated by an insulating material. The capacitance of such a device (if the plates are flat and parallel) is given by the equation:

$$C = (k \times e \times A) / d$$

where: C = capacitance (in farads)
e = a constant equal to 8.85×10^{-12} farad/meter
k = the dielectric constant of the material between the plates
A = the area of the plates (in square meters)
d = the distance between the plates (in meters)

You can experimentally measure the capacitance of homemade capacitors by connecting them across the T2 terminals of this project. One way of making a capacitor is to use two large sheets of aluminum foil separated by a thin insulating layer. A diagram of the capacitor is shown below:



A HOMEMADE CAPACITOR

One surprising thing about this experiment is how difficult it is to make a capacitor with even a measurable capacitance. You need to use very large metal plates (aluminum foil), and you need to keep the plate separation very small. Measurable capacitances can be obtained with plates as large as a piece of standard notebook paper, but bigger plates will yield better results. Try several different types of insulators in the space between the plates (paper, plastic, nylon, etc.). Some kinds of "plastic" have higher dielectric constants than others. Also try various thicknesses of materials. Refer to a college physics text for a table of dielectric constants for various materials (for example, *Physics* by David Halliday and Robert Resnick, New York: Wiley, 1966). Try to determine experimentally the dielectric constant for a particular material using the equation above. The most difficult thing to measure is the separation of the plates, which is the thickness of the insulator. Use a micrometer to measure this thickness, if one is available. Otherwise, do the best you can by measuring the thickness of the material folded over several times.

You can build homemade capacitors with much larger capacitance by "rolling up" the plates and the insulator.[5] Place a sheet of plastic or paper insulator on top of a slightly smaller sheet of aluminum foil. Place a second aluminum foil sheet, the same size as the first one, on top of the insulator. Roll the three sheets together around a paper tube. Make sure the two aluminum foil sheets do not touch and "short out." You can check this by using an ohmmeter between the two aluminum sheets. The resistance should be very high. To measure the capacitance, connect leads from each of the foil sheets to the T2 terminals.

◊ Temperature coefficients of capacitors and resistors

For many electronic circuits, it is essential that the capacitance of a particular capacitor remain constant. Unfortunately, capacitance is affected by temperature changes. This effect can be detected using this capacitance meter. Try a large-value (2 µF to 10 µF) electrolytic capacitor first. Electrolytic capacitors look like small metal cans. They have especially poor temperature stability. These capacitors are usually polarized, so make sure you orient the "+" lead properly. Place the capacitor in T2 and start the program reading the count (or a capacitance, if you have modified the program). Hold the capacitor in your fingers to warm it. You should notice a change in the reading. Try using a hair dryer to warm the capacitor slightly. If you have separated the two halves of the circuit board, you can do this without heating the built-in components. Try other types of capacitors: tantalum, ceramic, mylar, polystyrene, etc. Which types are the most temperature sensitive? Does the capacitance increase or decrease as you warm the capacitor? Estimate the temperature coefficient of each type of capacitor (% change per degree Celsius).

Similar experiments can be conducted on the temperature stability of resistors. As with capacitors, there are several types of resistors: carbon composition, metal film, wire wound. The various types differ in temperature stability. Temperature effects will be easiest to notice when using large resistance (1 MΩ or so) carbon-composition resistors. Does the resistance increase or decrease as the resistor is warmed? Which types of resistors are most sensitive to temperature change? Estimate the temperature coefficient of each type of resistor (% change per degree Celsius).

◊ Experiments with the resistance of soil

Use the circuit and software to measure the resistance of soil. Place two probe wires leading to terminal T1 into the ground a few centimeters apart. Is the resistance measurable? How is the resistance related to soil moisture content, type of soil, etc. Can you design a computerized system that will warn you when it is time to water your house plants?

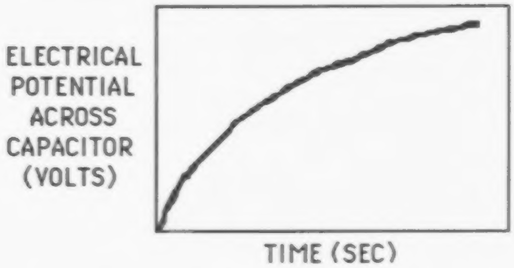
Additional Mathematics Background

A more complete explanation of the charging circuit and how its variables are related is given below:

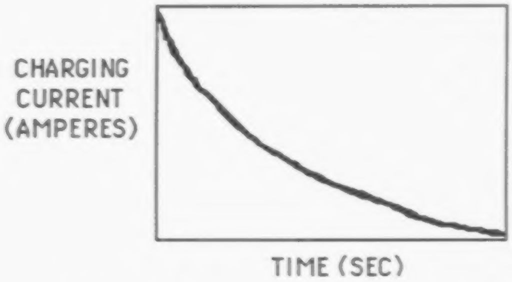
When a capacitor C is charged by a resistor R, the voltage across the capacitor increases; however, it does not increase at a constant rate. The graph of voltage versus time looks like this:

---

[5] This is the way many commercial capacitors are made.



The rate at which the capacitor charges decreases, because as the capacitor becomes charged the voltage difference between the two ends of the resistor decreases. The result is a reduced current, which in turn results in a slower charging rate. The graph of the charging current versus time is shown below:



Mathematically, these two relationships are stated in the two equations below:

$$V = V_0 (1 - e^{-t/RC})$$
$$= I_0 e^{-t/RC}$$

These are exponential equations. You can use these equations to calculate the amount of time it should take a specific capacitor to charge up to the threshold level in our system. The threshold level is 2/3 the +5V power-supply voltage, or about 3.3 volts. To determine how long it should take for the voltage to reach this level, we substitute into the top equation:

$$\begin{aligned} 2/3\, V_0 &= V_0 (1 - e^{-t/RC}) \\ 2/3 &= 1 - e^{-t/RC} \\ 1/3 &= e^{-t/RC} \end{aligned}$$

Taking the natural log of both sides of the equation:

$$\begin{aligned} \ln(1/3) &= -t/RC \\ -1.099 &= -\,t/RC \\ t &= 1.099\, RC \end{aligned}$$

Therefore, the time it should take to charge up the capacitor (in seconds) can easily be calculated if you know R and C. The resistance in this equation must be in ohms. The capacitance must be in farads. Notice that this equation is basically the same as the simple one we used in the early part of this project:

charging time = k x R x C

The charging time in terms of counts can be determined by remembering that each count takes 18 µsec. A sample calculation is shown below using the 47-kΩ resistor to charge the 0.022-µF internal capacitor only. How many counts should the program read (this is the situation when only switch S1 is closed and nothing is connected to terminal T1 or T2)?

$t = 1.099 \text{ R} \times \text{C}$
$t = 1.099 \times (47{,}000\ \Omega) \times (0.022 \times 10^{-6} \text{ F})$
$t = 1.14 \times 10^{-3} \text{ sec} = 1.14 \text{ msec}$

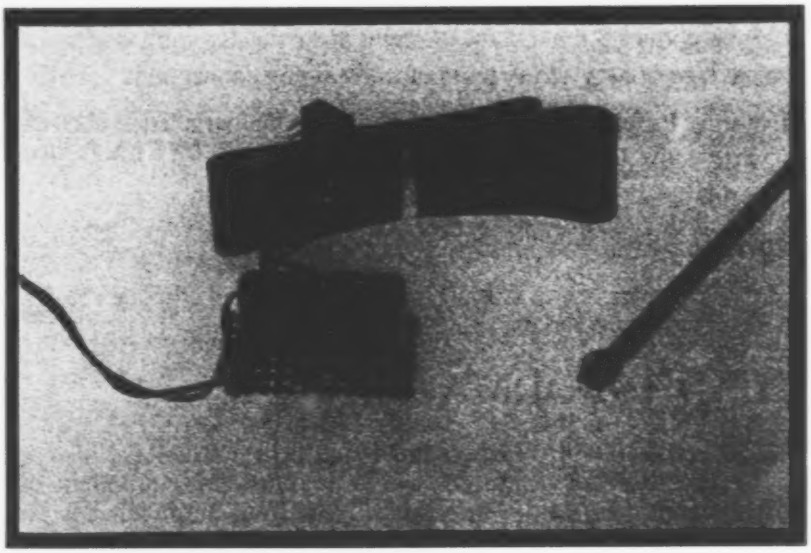
How many counts is this?

$(1.14 \times 10^{-3} \text{ sec}) / (18\ \mu\text{sec}) = (1.14 \times 10^{-3} \text{ sec}) / (18 \times 10^{-6} \text{ sec}) = 63$

This calculation shows that when the 47-kΩ resistor is used to charge just the 0.022-μF capacitor, the count reading should be about 63. Is this what you get when you try it? If not, either your resistor or the internal capacitor is different from the expected value.

The value of R x C is called the *time constant* of the circuit. It can be shown that the product of R x C works out to be the time it takes the capacitor to charge to 63% of its maximum voltage. RC time constants are often mentioned in describing electronic circuits.

PROJECT 6:
IC TEMPERATURE PROBE



This project consists of building a temperature-probe system that will accurately measure temperatures between -55°C and +150°C (-67°F to +302°F). The system uses an integrated-circuit temperature sensor known as an AD590. This temperature sensor produces a current output proportional to its absolute (kelvin) temperature. Programs on the *Project Programs* disk allow the computer to measure this current and display the temperature.

PROJECT REQUIREMENTS

Background Knowledge

- Electronics: To understand the operation of the temperature-probe system, you should be familiar with what capacitors are and what they do. You should also understand Ohm's law and the terms *charge*, *current* and *voltage*. The actual assembly of the circuit is fairly easy. Some experience with soldering would be helpful.
- Programming: None required for core project. Several of the project extensions require knowledge of BASIC.

Parts

- ☐ Analog Devices AD590JH Temperature Transducer (This component is not sold by electronic supply houses. It can be obtained directly from Analog Devices, Route 1 Industrial Park, P.O. Box 280, Norwood, MA 02062, (617) 329-4700. Unfortunately, Analog Devices has a fairly large minimum order. The "Supplemental Information" section of this chapter contains more information on this integrated circuit.)
- ☐ 4.7-µF capacitor, 10 volts or more (You can use almost any type of 4.7-µF capacitor, but some types of capacitors are much better than others in terms of temperature stability. The more stable this capacitor is, the more accurate your system will be. Here is a list of possible types of capacitors in order of preference, the best listed first: polycarbonate, polyester, Mylar, tantalum, electrolytic.)

- ☐ circuit board (Almost any small circuit board will work. One possibility is one-half of a Radio Shack #276-159.)
- ☐ cable to Apple game port (see *Appendix A*)
- ☐ glass or metal tube, approximately 15 cm long
- ☐ 2–conductor wire, 1 or 2 meters long (Any flexible, multi-stranded wire can be used. A 2–conductor wire with different color leads will make the assembly somewhat easier.)
- ☐ plastic electrical tape or heat-shrink tubing for insulating connections

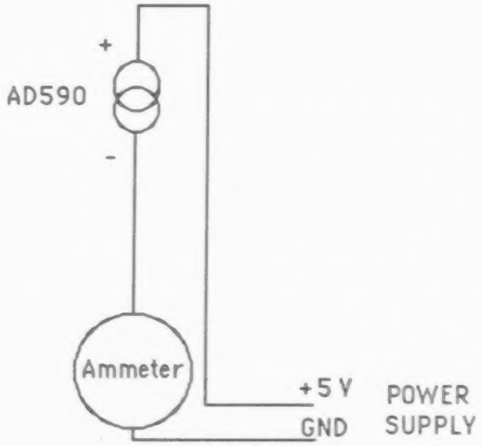
Note: An "IC Temperature Probe Parts Kit" containing all of the parts listed above (with a polyester capacitor and a brass probe tube) is available from Vernier Software for $13.00. An order form is in the back of this book.

Tools and Supplies

Soldering iron, solder, wire cutters, wire strippers, thermometer and silicone sealant (GE Silicone II or equivalent, available at hardware and discount stores).

BACKGROUND INFORMATION

The heart of this project is the AD590 Temperature Transducer.[1] The AD590 is a small integrated circuit (about the size of the tip of a match), which contains the equivalent of about a dozen transistors, six resistors, and a capacitor. It produces a current output that is proportional to its absolute temperature.[2] This current is ideally 1.0 microampere (1.0 x 10^{-6} ampere) per kelvin. A simple temperature-measurement circuit using the AD590 is shown below. Notice that only two leads to the temperature transducer are needed. Two overlapping circles is the symbol used on schematic diagrams for the AD590.



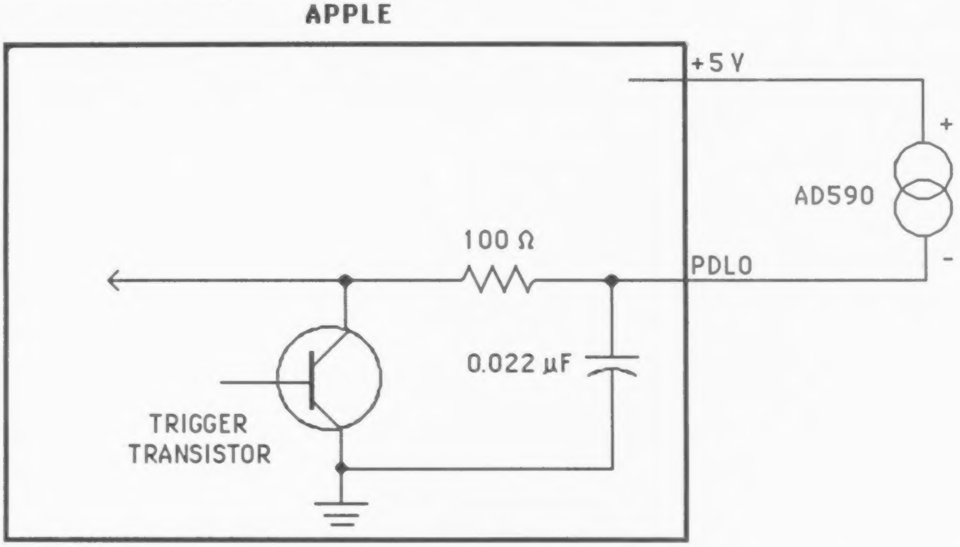
The current in microamperes measured by the ammeter is numerically equal to the temperature in kelvins. In this project, the Apple computer will be used in place of the ammeter in a circuit similar to this. The signal from the AD590 will come into the computer through the game port. The PDL0 line is used. The PDL lines are normally used for game paddles and joysticks. Their operation is explained in detail in "The Apple Game Port" section of the *Introduction*. We recommend that you take a few moments to read that section if you are not familiar with how the computer normally senses the condition of the PDL lines.

---

[1] A *transducer* is any device that produces an output signal in response to a stimulus.

[2] *Absolute temperature* refers to a temperature measured relative to absolute zero (the temperature at which all molecular motion ceases). Absolute temperatures are usually measured in kelvins. A temperature in kelvins is equal to the temperature in degrees Celsius plus 273.15.

One way of using the AD590 with the Apple is shown below. The components drawn inside the rectangle are inside the computer.



With this system, a current that is proportional to absolute temperature is supplied by the AD590. The current is used to charge the Apple's internal capacitor. The status of the PDL0 line is detected by measuring how long it takes the voltage across the 0.022-µF capacitor to rise from 0 to 3.3 volts after the trigger transistor is turned off. This is normally done with the Applesoft BASIC command PDL(0). Since the current flow is proportional to the absolute temperature of the AD590, the time required to charge the capacitor is directly affected by the temperature of the AD590. If the AD590 is cold, it produces a small current, so charging the capacitor takes a long time. If the AD590 is warm, it produces more current and the capacitor charges more quickly.

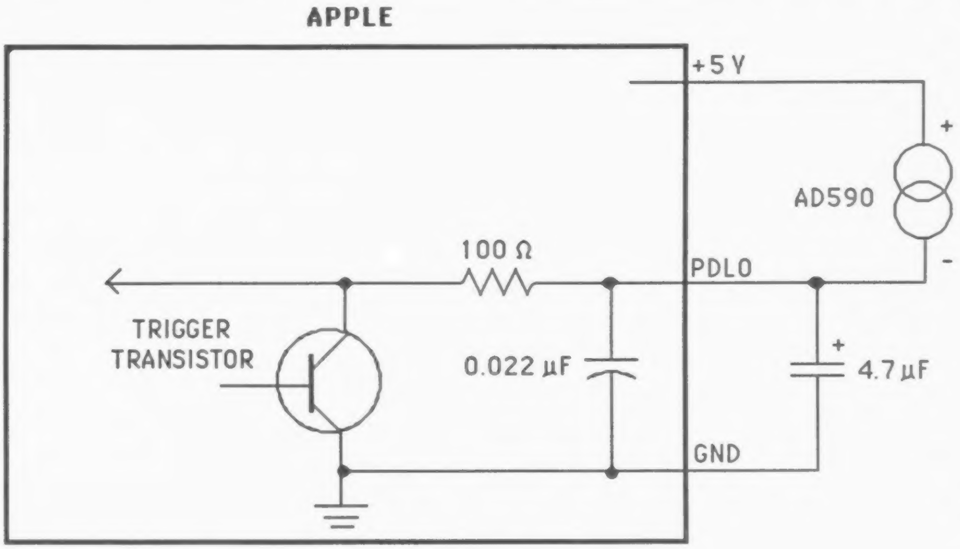
The computer counts how many cycles of a timing loop occur before the voltage across the 0.022-µF capacitor reaches 3.3 volts. The count reading registered by the computer is a measure of the charging time, and is inversely proportional to the current. The reciprocal of the count reading, "1/counts" is thus a measure of the current. Since the current is proportional to the absolute temperature of the AD590, the relationship between counts and temperature is:

$$T = k \times (1/\text{counts})$$

where k is a proportionality constant that depends on the time units and the electrical values in the circuit. When the Applesoft BASIC command PDL(0) is used to measure the charging time, an integer "count" ranging from 0 to 255 is reported. Each count represents 11 microseconds of charging time.

The circuit diagrammed above can be used with the PDL(0) command to form a simple electronic thermometer; however, it would not be very satisfactory. At room temperature (20°C) it reads 23 counts. In boiling water (100°C) it reads 18 counts. The count cannot vary through enough range to provide much accuracy. A better system is needed if this electronic thermometer is to be practical.

We will make a hardware and a software change to improve the situation. The improved circuit is shown on the schematic below:



THE IC TEMPERATURE PROBE SYSTEM

A 4.7-µF capacitor is added to the circuit. This extra capacitor greatly increases the amount of charge that must be delivered by the AD590 to cause both capacitors to reach the threshold voltage level. It will take the transducer a much longer time to deliver that charge. In fact, it takes so long that the Apple's PDL(0) command can no longer be used to read the charging time. The PDL(0) command stops counting at 255, which represents about 3 msec. We will use a special machine-language program called PDL.CHECK.[3] This routine reads the time it takes the capacitor to charge in a similar way, but it can count for a much longer time. The PDL.CHECK program counts from 0 to 40,960, with each count representing 18 microseconds. It can measure charging times of up to 0.7 seconds, which is plenty of time for the AD590 to charge the capacitors. With this new timing routine and the extra capacitor, the count will range from about 3000 at 20°C to about 2000 at 100°C. There are about 1000 possible count readings spread over these 80°C, so this system can read temperature down to a fraction of a degree Celsius.

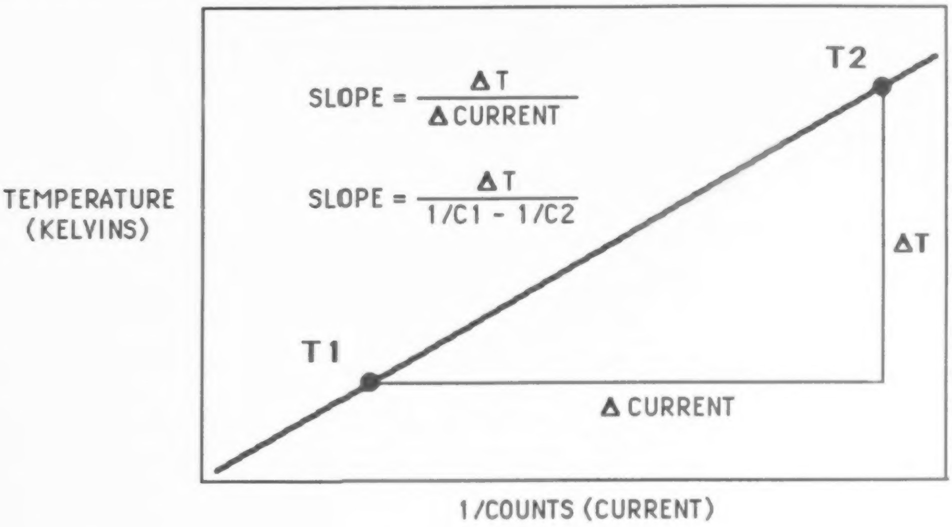
The accuracy of this system depends on the capacitance of the 4.7-µF capacitor remaining constant. Certain types of capacitors are much better in terms of temperature stability than others. Specific recommendations of capacitors are made in the parts list.

The probe can easily be calibrated to read temperatures and not just counts. Since the quantity "1/counts" is proportional to absolute temperature, reading the count value while the AD590 is at two known temperatures (measured with a thermometer) provides the necessary calibration data.[4] Assume that the AD590 was placed in a water bath at a temperature T1 and it reads C1 counts. Then it is placed in a second, warmer water bath at temperature T2 and it reads C2 counts. The calibration is based on the slope of the straight line between the two calibration points as shown in the figure below:

---

[3] The PDL.CHECK program is also used in Project 5, *Resistance/Capacitance Meter*.

[4] Actually, you can calibrate using only one known temperature if you assume the current at zero kelvins would be zero microamperes, but accuracy is improved by doing the calibration at two different measured temperatures.



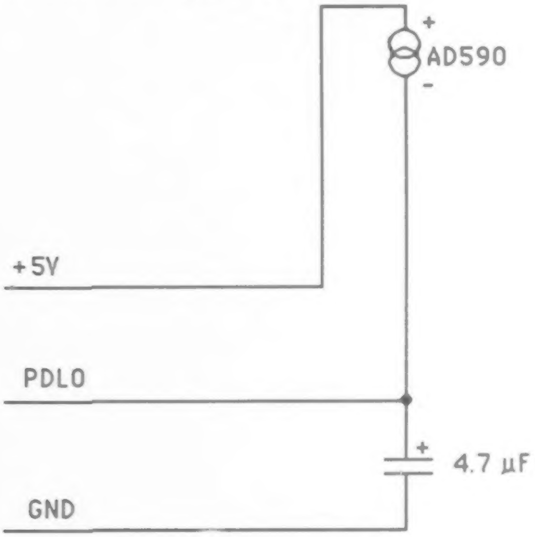
The slope of this line is the change in temperature divided by the change in "1/counts". Once the slope is calculated, count readings from the computer can be converted into temperatures using the formula:

temperature = slope x (1/counts - 1/C1) + T1

The program TEMP.SENSOR on the *Project Programs* disk includes a mode for calibrating the sensor and another mode for displaying the temperature reading. Many of the suggested project extensions involve adding new features to this program.

STEP-BY-STEP INSTRUCTIONS

The complete schematic for the temperature-probe system is shown below:



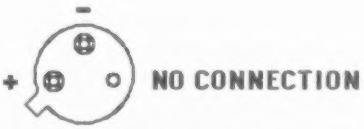
SCHEMATIC OF IC TEMPERATURE PROBE

☐ 1. **Solder the capacitor to the circuit board.** Solder the 4.7-µF capacitor to the circuit board. If the capacitor you are using is polarized (one side is marked with a "+" sign), make note of which are the "+" and "-" leads.[5]

☐ 2. **Prepare the cable to the Apple game port**. Refer to *Appendix A* for information on cables. Only the following leads will be used: +5V, PDL0 and GND. Strip a few millimeters of insulation from the ends of these three leads.

☐ 3. **Connect the cable leads to the circuit board**. Solder the PDL0 line of the game-port cable to the "+" lead of the capacitor (or either lead, if your capacitor is not polarized). Solder the GND line to the other lead of the capacitor. Solder the +5V line to an isolated spot on the circuit board. Later, one of the leads from the temperature probe will connect to this spot.

☐ 4. **Solder a two-conductor wire to the leads of the AD590**. The AD590 can be damaged by static electrical discharges. Remove any electrical charge that may have built up on your body by touching a grounded metal object (like a water pipe) with your fingers. Now pick up the AD590, and identify the leads. The diagram below shows the leads for the standard (H package) AD590.



AD590 H PACKAGE (BOTTOM VIEW)

Notice that the diagram is a bottom view (leads pointing at you). There are three leads, but you will not use the lead that is directly connected to the metal housing of the integrated circuit. Cut it off or at least make sure that it is not allowed to touch the other leads when the probe is completed. Notice that the positive lead is the one closest to the tab.

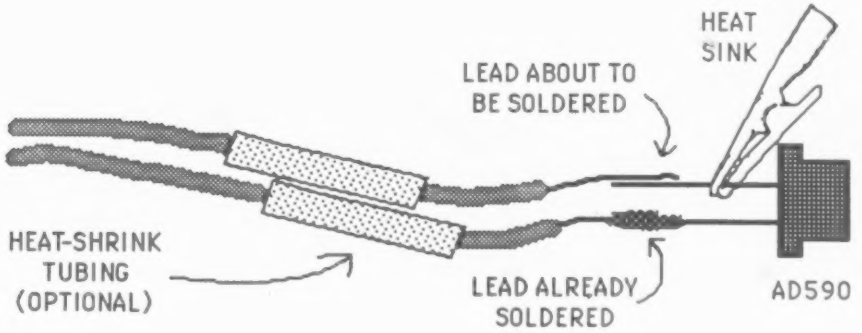
Heat-shrink tubing is recommended for insulating the connections between the wire and the AD590 leads. If you are using heat-shrink tubing, place the tubing over one of the wire leads before you start to solder. If you do not have heat-shrink tubing, you can use plastic electrical tape to insulate the connections after they are soldered.

Place a heat sink (alligator clip or the jaws of a pair of pliers) on the lead of the AD590 before soldering (see diagram below). This minimizes the flow of heat from the solder joint into the temperature sensor. The AD590 can be damaged if overheated.

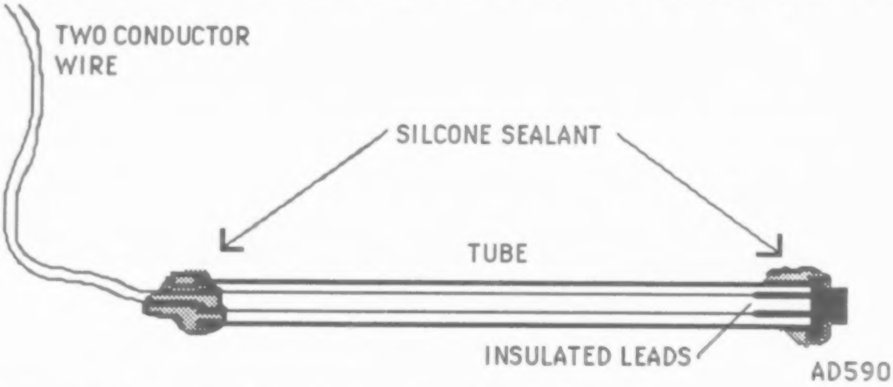
Strip a centimeter of insulation from one of the wires and solder it to one of the AD590 leads. Remember that the soldered leads from the temperature transducer will have to fit inside the probe tube, so keep the junction compact. Change the heat sink to the other lead, and attach it in a similar manner. Insulate the two leads from each other by moving the heat-shrink tubing into place or by using plastic electrical tape. **Note**: If you use a heat source to shrink the heat-shrink tubing around the leads of the AD590, avoid overheating the body of the AD590.

---

[5] The capacitor included in the Vernier Software "IC Temperature Probe Parts Kit" is not polarized.



☐ 5. **Complete the temperature probe.** Mark the wire that connects to the "+" lead of the AD590.[6] Later you will need to know which wire is positive. Slip the 2-conductor wire through the tube. Pull the temperature transducer up to the end of the tube. Use a silicone sealant to glue and seal the device to the end of the tube. It is very important to get a watertight seal around the AD590. The probe will not read properly if water gets inside. However, do not get sealant on the flat end of the AD590. Sealant there will insulate the device from the environment and slow down its response to temperature change. Remember that the choice of sealant used here affects the chemicals in which the temperature probe can be used. We have found Silicone II from General Electric to be very durable.

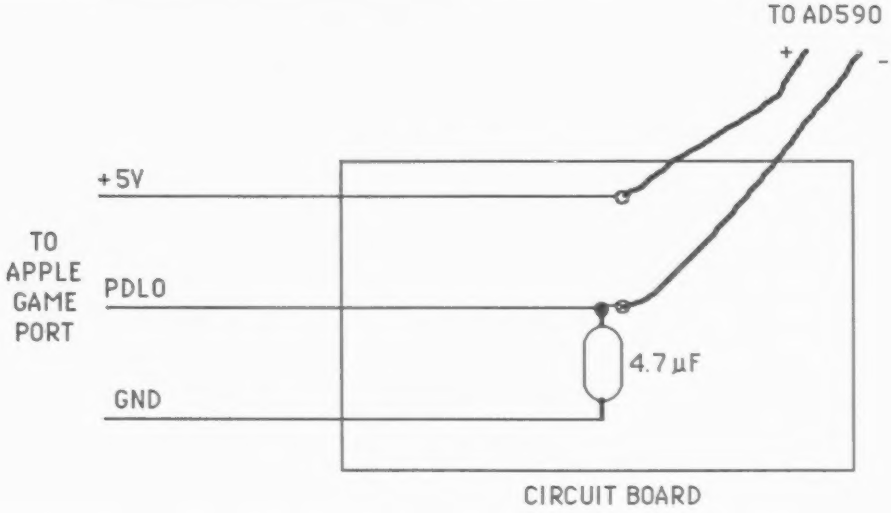


Place sealant at the other end of the tube as well. This serves two purposes: (1) It prevents solutions from getting down the tube if the probe is totally submerged; and (2) It reduces the chances of the wires wearing out where they bend at the end of the tube. Let the sealant dry thoroughly before handling the probe again.

☐ 6. **Connect the leads from the temperature probe to the circuit board.** The lead that is connected to the "+" terminal of the AD590 connects to the +5V lead from the computer. The other lead connects to the PDL0 lead from the computer (which is also connected to the "+" side of the capacitor). Strip the insulation from these wires and solder them to the circuit board. Refer to the diagram below:

---

[6] If the 2-conductor wire you are using has leads of different colors, just remember the color of the "+" lead.



☐ 7. **Test the IC temperature-probe system**. Turn off the computer and connect the circuit to the game port. Place the *Project Programs* disk in the disk drive and turn on the computer. If the computer acts strangely or will not turn on, flip the power switch off and double-check your circuit—you must have an error in wiring, probably involving the +5V line.

If the disk starts up normally, run the TEMP.SENSOR program. When the menu is presented, choose the "DISPLAY COUNTS" (C) mode. The computer should begin to display the count reading. It should be between 3500 and 2500. Touch and warm the AD590 and see if the count changes. The count should get lower as the temperature increases. If these tests are normal, you have a working probe system and you are ready for calibration. If the probe is not working properly, refer to the "Troubleshooting" section below.

☐ 8. **Calibrate the probe**. Select the "CALIBRATION" (Z) mode from the menu of the TEMP.SENSOR program. Place the probe in a container of cool water. Stir the water and measure the temperature of the water in °C with a good thermometer. Leave the probe and the thermometer in the bath long enough for them to reach a steady reading. Press the <RETURN> key and then enter the actual temperature in °C. Repeat this process in hot water. The program will calculate the value of the slope and saves the calibration values on the disk in a file called TEMP.CAL. From this time on, whenever the TEMP.SENSOR program is used, these calibration values will be loaded into memory so that the program can calculate actual temperatures from the count readings. If you ever want to recalibrate, simply select the "CALIBRATION" mode again.

☐ 9. **Check the accuracy of the finished IC temperature probe**. Select "DISPLAY TEMPERATURE" (T) from the menu of the TEMP.SENSOR program. The program should continuously display the count and the temperature reading in degrees Celsius.[7] Check the accuracy of the temperature probe at ten different temperatures over the range 0°C to 100°C. Compare the readings from the probe with thermometer readings. Is the temperature probe accurate over the entire temperature range? Determine the maximum error (that is, the largest difference in readings between the probe and your thermometer). Check to see if the probe readings are reproducible; that is, if the probe is moved out of and back into the same temperature several times, does it read the same value? Refer to the "Supplemental Information" section below for tips on using the temperature probe.

---

[7] Notice that too many digits are used when the temperature is displayed. The temperature should probably be displayed to the nearest 0.1 degree. Improving this part of the program is left as one of the project extensions.

TROUBLESHOOTING

Check the following if your temperature probe system is not working properly:

◊ If the computer will not start up, there must be a short circuit (direct connection between the +5V and the GND leads).

◊ If the count displayed by the "DISPLAY COUNTS" mode is -24576, either the probe leads are not connected or the AD590 leads are reversed.[8]

◊ If the count displayed by the "DISPLAY COUNTS" mode is 1, the probe is shorted. Check for a connection between the AD590 leads inside the probe.

◊ If a probe reading changes inconsistently, check for water inside the probe.

If none of this helps:

◊ Check to make sure that you are using the leads from the Apple game port properly. Use an ohmmeter to check the wiring from the pins that go into the computer to the circuit board. Are you using the PDL0 line and not some other one?

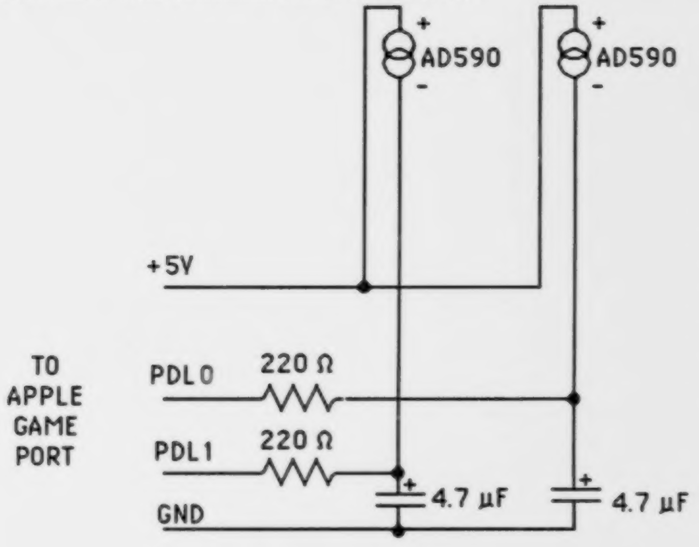
◊ If you are using a 16-pin game-port cable, is it inserted with pin #1 in the right spot? Are all 16 pins inserted in the socket?

◊ If you used a polarized capacitor, make sure the capacitor is oriented properly. The "+" lead should be connected to PDL0.

PROJECT EXTENSIONS

Circuit Modifications

◊ Add a second temperature probe to the system. An additional temperature probe can be added to the circuit using the PDL1 line. The schematic is shown below:



TWO TEMPERATURE PROBE SYSTEM

This circuit is essentially a double version of the circuit described earlier. The two extra components are the 220-Ω resistors added to the PDL input lines. These resistors are not essential, but they help minimize interaction between the two probes. To use both probes, you will have to use the TEMP.SENSOR.2 program that is included on the *Project Programs* disk. This program is similar to the TEMP.SENSOR program, except that it reads both probes.

---

[8] Operating the AD590 with the leads reversed does not permanently damage it.

- One of the nice things about measuring temperatures with a current-source temperature transducer is that long lead wires can be used without affecting the accuracy of the system. Try adding several meters of wire between the circuit board and the temperature probe. Check how much the calibration and accuracy of the system are affected by this change.

Program Improvements

There are a number of improvements you might want to make to the TEMP.SENSOR program. The "Program Design Notes" section below explains how the TEMP.SENSOR program operates. It should help you in making modifications. Some suggested changes are given below:

- Round the temperatures displayed to the nearest 0.1°C.
- Change the program so it can collect data in memory and display it on the monitor or print it later.
- Change the program so that it displays the temperatures in degrees Fahrenheit, degrees Celsius, or kelvin.
- Allow the temperature data to be saved on a disk.
- Have the program display the time of each temperature reading measured from the start of the readings. Notice that each temperature measurement takes about one second.
- If your computer has a "clock card" to keep track of real time, change the program so that it reads the clock time and stores it along with the temperature reading. This would be very useful for a weather station.
- Change the program so that temperature readings at preselected time intervals (for example, every hour) are saved in memory. Again, this would be desirable for a weather station.
- Program the computer to record the maximum and minimum temperature reading.
- Change the program so that the computer makes a warning noise whenever the temperature is outside a specified range.
- Add a mode to the program that graphs the temperature readings versus time.
- Add a mode to the program that displays the temperatures in large, block digits so that they can be read at a distance.
- If you have modified the circuit to allow the use of two probes, change the TEMP.SENSOR.2 program so that it can be used as a humidity meter. Cover one probe with a wet cloth and leave the other one dry. Have the computer read the two temperatures, calculate the difference, and then determine the relative humidity.

Additional Experiments

The IC temperature probe can be used for a number of experiments:

- Plotting cooling curves
- Measuring specific heats and heat capacity
- Measuring the latent heat of fusion of ice
- Testing for changes in the freezing point of water when various chemicals are added
- Studying temperature at various heights in a room
- Measuring rates of cooling and the efficiency of insulation
- Measuring heat of reaction
- Studying outdoor temperature variation
- Studying indoor temperature, thermostat function, furnace cycles, etc.
- Studying thermal gradients in a container of water

SUPPLEMENTAL INFORMATION

Suggestions for Using the Temperature Probes

Calibration - Careful calibration of the probes is important for accurate temperature readings. Always use a good thermometer. The accuracy of the probes can be only as good as the device with which you calibrate.

Equilibrium - The most common cause of problems in calibration is not allowing the probes (or the thermometer) to reach equilibrium with the water bath. Stirring the water bath is a good idea to make sure the probes and the thermometer are actually at the same temperature.

Choice of calibration temperatures - The output from the AD590 is theoretically a current proportional to its absolute (kelvin) temperature. It does a very good job at this, but of course, there is some deviation from perfect linearity. The probe will read most accurately if it is calibrated at temperatures similar to the temperatures to be encountered. For example, if you are going to use the probe to measure indoor temperatures, it would be best to calibrate it at 15°C and 30°C instead of at 0°C and 100°C.

Using probes in chemicals - Be careful when using the temperature probes in chemicals. Some chemicals may react with the temperature transducer, probe tube, or with the sealant. Cyclohexane and other nonpolar solvents will dissolve silicone sealant.

Wet probes - If you start getting strange results from a probe, a likely cause is liquid (usually water) inside the probe tube. You can test for this by shaking and tapping the probe in the air after you have dried the outside. The probe reading should not be affected by these vibrations. If the reading changes by a degree or so, it is likely that the probe is wet inside and that the trapped liquid is causing problems. Take the probe apart (use a knife to cut off the silicone sealant), dry it out and reseal it.

Calibration change due to changes in room temperature - If you are using capacitors that have poor temperature stability (for example, electrolytic capacitors), you may need to recalibrate if the temperature of the room in which you are working changes significantly. This becomes necessary if the capacitance of the 4.7-µF capacitor changes.

Additional Information on the AD590

We use the AD590 as a temperature sensor instead of other possible devices (such as thermistors, diodes, etc.) for two reasons: (1) Its response is directly proportional to absolute temperature, which makes the device easy to calibrate; and (2) A temperature sensor that produces a current output (instead of a voltage output) does not have its reading affected by the resistance of the wires leading from the sensor to the readout device. With the AD590, very long wires can be used from the temperature probe to the computer (as you might want for a weather station).

The AD590 is sold in two different packages. The "H" package is a metal can that looks very much like a metal transistor. This is the package recommended for this project and discussed in the "Step-by-Step Instructions." The "F" package is a much smaller, flat ceramic device with only two leads. Transducers in this package are more expensive, but since they are less massive, they respond to temperature changes faster. The package ("H" or "F") is identified by the last letter in the Analog Devices part number.

Analog Devices tests all of the AD590 temperature transducers and sorts them according to how well they match their theoretical output (1.0 µamp/K). The devices are then sold in various grades: I, J, K, L, and M. The grade is indicated by the second to the last letter of the part number. Grade I devices are the cheapest and least linear; grade M transducers are very linear and expensive. We have found that grade J transducers give an accuracy of within a few tenths of a degree Celsius. The AD590JH transducers cost about $5.00 if purchased individually.[9]

Program Design Notes

The TEMP.SENSOR program is an Applesoft BASIC program that can determine the count reading from the PDL0 input line and convert that count into a temperature. It also includes a calibration routine that leads the user through the necessary operations to calibrate the probe. The program then calculates the calibration constants and stores them on the disk. Each mode of operation is explained below:

DISPLAY COUNTS - The count reading is actually determined by the machine-language program PDL.CHECK.[10] This program is loaded into memory starting at memory location 768 ($300 in hexadecimal notation) when the TEMP.SENSOR program starts up. Whenever the status of the PDL line is to be checked in the DISPLAY COUNTS mode, the BASIC program executes this statement:

```
CALL 768
```

This statement causes the PDL.CHECK machine-language program to begin operation. The charging time for the capacitor (in counts) is measured. This takes about one second. At the end of the PDL.CHECK program, the count reading is stored in two bytes of memory:

974: low byte 975: high byte

When the PDL.CHECK program is completed, the BASIC program continues. It determines the count reading by checking the two bytes and calculating the counts as a decimal number using this statement:[11]

```
CT = PEEK(974) + 256 x (PEEK(975) - 96))
```

The value of CT is then displayed on the screen as the count reading.

The program will continue displaying counts and temperatures until a key is pressed. It then displays the menu again.

CALIBRATION - In this routine, the count and temperature at two different temperatures are entered into the computer. First the program instructs the operator to place the probe and a thermometer in a cool water bath. The program begins reading the count continuously. When the thermometer and the probe have reached equilibrium, the operator presses <RETURN>. This causes the computer to take the last count reading as C1, which is used in calibration. The actual temperature, T1, is now entered. The same procedure is then repeated to enter C2 and T2 into the computer. These four values are now stored on the disk in a sequential file called TEMP.CAL. The file consists only of the four values: C1, T1, C2, and T2. Whenever the main TEMP.SENSOR program begins, this file is read from the disk so that the proper conversion from counts to temperature can be made.

---

[9] For more information about the AD590, refer to any of the following publications from Analog Devices: "Two-Terminal IC Temperature Transducer" data sheet, "Application Note: Accuracies of the AD590," and *Transducer Interfacing Handbook* edited by Daniel H. Sheingold.

[10] The assembly and machine-language lists of the PDL.CHECK program are in *Appendix E*.

[11] Because of the way the machine-language program operates, the count total is stored in a complicated way. The number in the low byte represents a number of counts. The number in the high byte starts at 96 and increases by one for every 256 counts registered.

DISPLAY TEMPERATURE - This mode will check and display the count exactly as the "DISPLAY COUNTS" mode does. It also converts counts to temperature by using the following statements:

SLOPE = (T2 - T1) / (1/C2 - 1/C1)
T = SLOPE (1/CT - 1/C1) + T1

T is the temperature when the count reading is CT.

The program will continue displaying counts and temperatures until a key is pressed.

The following variables are used in the TEMP.SENSOR program:

C1, C2	COUNT READINGS AT LOW AND HIGH CALIBRATION TEMPERATURES
T1, T2	TEMPERATURES AT LOW AND HIGH CALIBRATION TEMPERATURES
CT	LATEST COUNT READING
T	TEMPERATURE
SL	SLOPE OF THE CALIBRATION LINE
M$	MODE OF OPERATION
D$	CONTROL D (USED FOR DISK ACCESS)

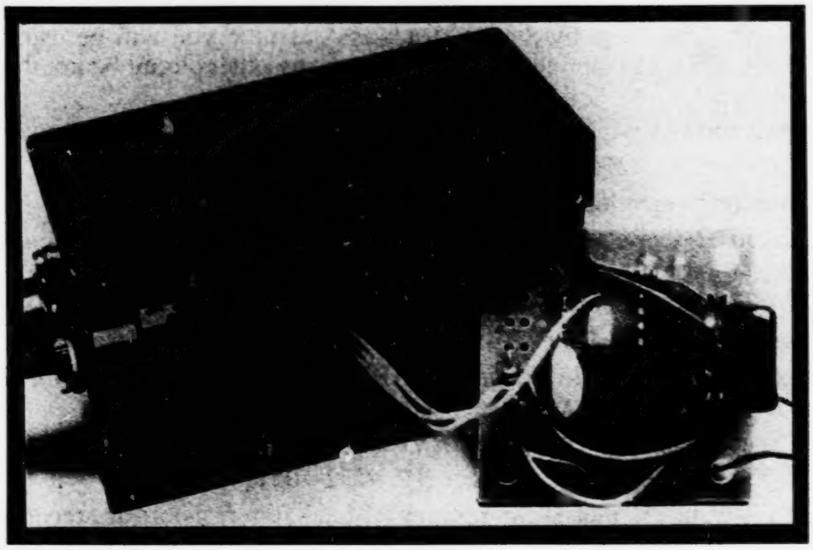
The two-probe version of the program (called TEMP.SENSOR.2) is very similar. Reading a second probe that is connected to the PDL1 line is done with the PDL.CHECK program by using a different CALL statement:

```
CALL 794
```

The count total is placed in the same memory location as before. The count can be checked using a "PEEK" command in BASIC as before. The following additional variables are used in the TEMP.SENSOR.2:

C3, C4	COUNT READINGS FROM PDL1 AT LOW AND HIGH TEMPERATURES
T3, T4	TEMPERATURES AT LOW AND HIGH CALIBRATION TEMPERATURES
CZ	LATEST COUNT FROM PDL1
S0	SLOPE OF PDL0 CALIBRATION LINE
S1	SLOPE OF PDL1 CALIBRATION LINE
T	TEMPERATURE FROM PDL0
TZ	TEMPERATURE FROM PDL1

PROJECT 7:
VOLTAGE MONITOR



There are many situations in which it would be useful to have a computer measure a voltage. Some experiments involve studying voltage changes directly. But more importantly, many devices are available that produce a voltage output as they measure something else. Examples include: pH meters, thermocouples, spectrophotometers and strain gages. Unfortunately, measuring voltages with the Apple computer requires a little extra hardware. The standard method of reading voltages with a computer is to use an *analog-to-digital converter*. This is a complicated, expensive piece of hardware that converts voltage signals into numbers that the computer can interpret.

In this project we will build an inexpensive device that will allow us to measure voltages through the Apple game port. We will use an integrated circuit called a *voltage-to-frequency converter*. This IC senses the voltage applied to it and then produces an output signal that varies in frequency with the input voltage. The computer can determine the frequency coming from the voltage-to-frequency converter and therefore indirectly measure voltage. Several simple projects are suggested to demonstrate the use of this voltage-monitoring system. There are also three other projects in this book that use the voltage-monitoring system: *Thermocouple* (Project 8), *pH Meter* (Project 9) and *Strain Gage* (Project 10).

PROJECT REQUIREMENTS

Background Knowledge

- Electronics: Experience with soldering components on a circuit board is recommended. The placement of the components on the board is clearly specified in the instructions.
- Programming: The core project requires no programming. Some of the project extensions require familiarity with BASIC.

Parts

- ☐ LM331 Precision Voltage-to-Frequency Converter (manufactured by National Semiconductor, Inc. and sold through many electronic-parts distributors)
- ☐ cable to Apple game port (refer to *Appendix A*)
- ☐ circuit board (The "Step-by-Step Instructions" assume you will be using one-half of a Radio Shack #276-159 circuit board. If you use any other circuit board, the assembly will be more difficult.)
- ☐ experimenter box (Any plastic or metal box large enough for the circuit board will work.)
- ☐ 8-pin IC socket (RS #276-1995)
- ☐ 20-kΩ multiturn potentiometer (RS #271-340)
- ☐ 5-conductor DIN chassis-mount socket (RS #274-005)
- ☐ cable with 5-pin DIN plug (one-half of a DIN patch cord [RS #42-2151] or a 5-pin DIN plug [RS #274-003] with two wires soldered onto it)
- ☐(2) test clips (RS #270-370, RS #270-372 or #270-336)
- ☐ 1-μF, 10-volt, metal-film capacitor (RS #272-1055) (Metal-film capacitors are recommended for this project because of their stability.)
- ☐ 0.1μF, 10-volt, metal-film capacitor (RS #272-1053)
- ☐ 0.01-μF, 10-volt, metal-film capacitor (RS #272-1051)
- ☐(2) 100-kΩ, 1/4-watt, 1% metal-film resistors (Metal-film resistors are recommended for the 100-kΩ and 6.8-kΩ resistors in this project. If you do not have access to metal-film resistors, you can use standard carbon-composition (5%) resistors. Carbon-composition resistors are not as stable, and the calibration of the final unit may not be quite as consistent as it will be if metal-film resistors are used.)
- ☐ 10-kΩ, 1/4 watt, 5% resistor
- ☐ 6.8-kΩ, 1/4 watt, 1% metal-film resistor
- ☐ 47-Ω, 1/4 watt, 5% resistor
- ☐ miscellaneous wire for connections on the circuit board

Note: A "Voltage-Monitor Parts Kit" is available from Vernier Software containing all of the components needed for this project, including metal-film resistors and capacitors, a 16–conductor ribbon cable to the Apple game port and an experimenter box. This kit sells for $24.00. An order form is provided at the back of this book.

Tools and Supplies

Soldering iron, solder, wire cutters, wire strippers, screwdriver, drill and drill bits.

BACKGROUND INFORMATION

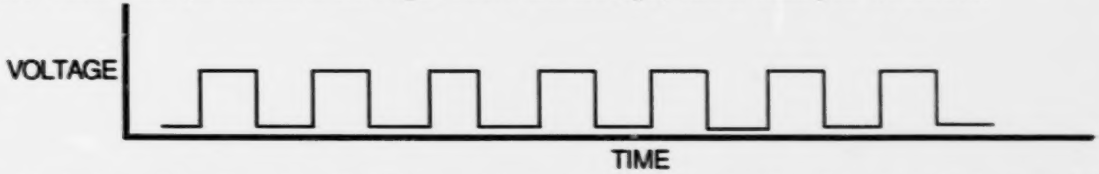
A voltage[1] that varies over a specific range is an example of an *analog* signal. An analog signal has an infinite number of possible values. Most computers (including the Apple II+ and the IIe) are unable to interpret such a signal directly. Extra hardware must be used to allow the computer to monitor an analog signal. The standard method of doing this is to use an *analog-to-digital converter*. There are a number of these devices available for the Apple II+ or IIe. They are sold as circuit boards that plug into one of the slots inside the computer. They cost from $100 to $600, depending on the precision to which they can measure voltage, how long they take for each measurement and how many different voltages they can monitor. If you are going to frequently use your computer for voltage measurement, an analog-to-digital converter board may be a good investment. Analog-to-digital converters are essential when you need to monitor voltages many times per second or when several voltages are to be monitored at once.

---

[1] In this book, and in almost all electronics books, the term *voltage* is used in place of the more scientific term *electrical potential difference*.

This project is an inexpensive alternative to an analog-to-digital converter board. It will let you measure voltage through the game port. The cost of the parts is approximately $24.00. After it is completed, this voltage-monitoring system can be used for a number of experiments. You will also need this system to do Projects 8, 9 and 10 of this book.

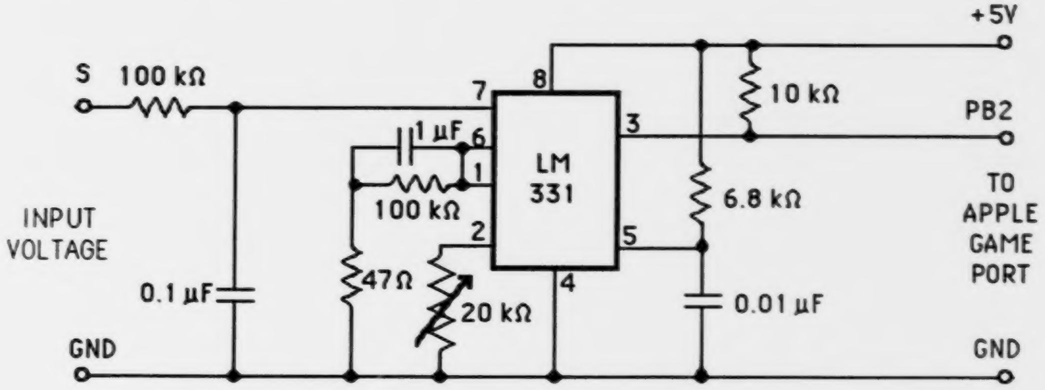
In this project we will use an integrated circuit called a *voltage-to-frequency converter*, which produces an output signal that the computer can interpret. If you graphed the output voltage of this device versus time (or looked at its output on an oscilloscope) this is what you would see:



This kind of output voltage is called a *square wave*. The *frequency* of such a wave is the number of voltage pulses produced per second. The frequency is usually reported in cycles per seconds or *hertz*. For example, if 500 square-wave pulses are sent out each second, the frequency is 500 Hz. If a square-wave signal is connected to one of the Apple game port's switch input lines, the computer can sense the voltage pulses in the same way that it can sense when a joystick button is being pressed. The computer can count the number of these pulses that occur in a specific time interval and then calculate the number of pulses per second (the frequency).

The voltage-to-frequency converter is specifically designed to produce a square-wave output with a frequency that varies in direct proportion to the voltage applied to it. This will allow us to indirectly measure voltage by having the computer determine the frequency of the square wave fed into the game port.

The complete circuit for this project is shown below:

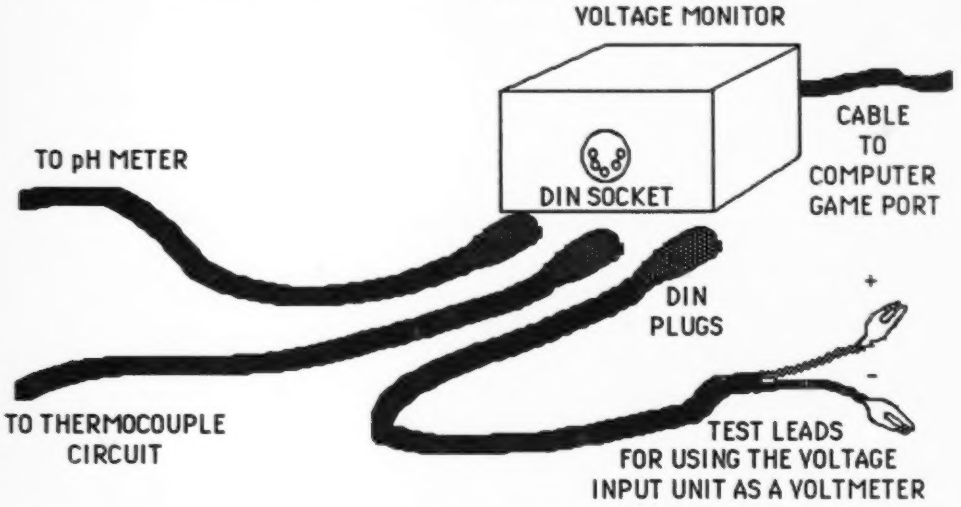


SCHEMATIC DIAGRAM OF THE VOLTAGE MONITOR

The heart of this circuit is the LM331 Precision Voltage-to-Frequency Converter IC. This is an integrated circuit that is manufactured by National Semiconductor, Inc. It is a complex circuit consisting of dozens of transistors, resistors and diodes. As is often done in modern electronics, we will treat the LM331 as a functional unit and not worry about how it does what it does.[2] The voltage to be measured is applied to pin #7 of the LM331. The voltage should be within the range 0 to 3.5 volts. The square-wave output is produced at pin #3. The potentiometer connected to pin #2 allows you to set the ratio of frequency to voltage. We will set this potentiometer so that the frequency output (in hertz) is equal to 1000 times the voltage input (in volts). The other resistors and capacitors are needed to set up the LM331 for accurate voltage-to-frequency conversion.

Once this circuit is assembled and connected to the Apple game port, the BASIC program VOLTAGE on the *Project Programs* disk will allow you to measure voltages with the computer. This program uses a machine-language routine to count the voltage pulses received at the switch input of the Apple game port.[3] It always counts pulses over a time interval of 0.501563 seconds. The number of pulses counted is divided by the 0.501563 seconds to yield the frequency of the wave being received. This frequency is divided by 1000 to yield the voltage. The program then displays the voltage.[4]

We suggest placing the circuit in an experimenter box and mounting a 5-conductor DIN socket in the side of the box. The Voltage Monitor can then be used in several different ways. If you connect a simple 2-conductor cable with test leads on the ends, it can be used for direct voltage measurements (like a voltmeter). Alternatively, you can connect other electronic circuits and study their voltage output. Projects 8, 9 and 10 are designed to be used in this way. They show how to build circuits to connect to the Voltage Monitor in order to indirectly measure temperature, pH and force.


VOLTAGE MONITOR
TO pH METER
DIN SOCKET
CABLE TO COMPUTER GAME PORT
DIN PLUGS
+
-
TO THERMOCOUPLE CIRCUIT
TEST LEADS FOR USING THE VOLTAGE INPUT UNIT AS A VOLTMETER

The project extensions include some circuit improvements, program enhancements and suggestions for other experiments that can be done using the Voltage Monitor.

---

[2] If you want more information about the LM331, refer to the *Linear Databook*, published by National Semiconductor. It includes eleven pages of information on how the LM331 operates, its specifications, and how it can be used.

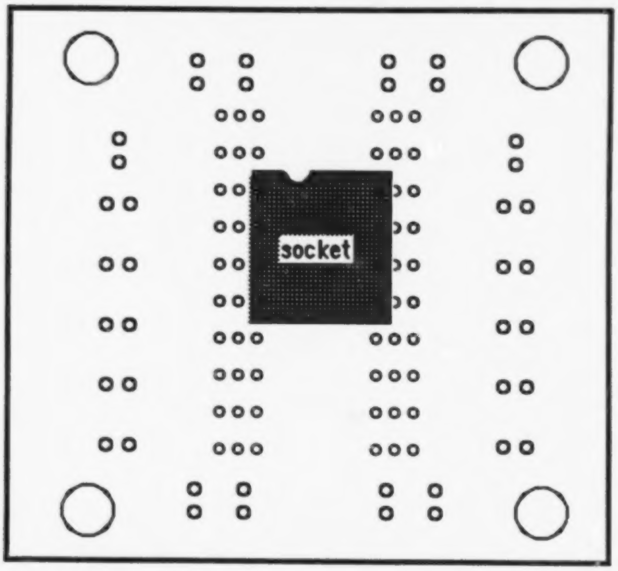
[3] In fact, the program VOLTAGE uses the FREQ program, which is also used to determine the frequency of input signals in Project 3 (*Frequency Measurements*) and Project 4 (*Humidity Meter*).

[4] Since an integer number of pulses is always counted, and the sample time is approximately 0.5 seconds, the smallest frequency change that can be detected by the computer is approximately 2 hertz. Thus the smallest voltage change that can be detected is 0.002 volts.

STEP-BY-STEP INSTRUCTIONS

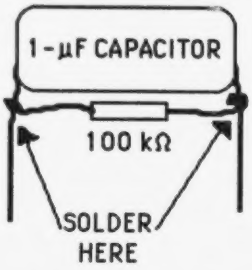
Note: These instructions assume that you are using one-half of a Radio Shack #276-159 Dual IC Circuit Board for mounting the components. If you are using any other circuit board or a prototyping board for assembling this circuit you will have to lay out the circuit yourself. Use the schematic diagram in the "Background Information" section as a wiring guide.

☐ 1. **Place the 8-pin socket on the circuit board.** Hold the circuit board (one-half of a RS #276-159) with the copper-plated side down and the words "Radio Shack" at the bottom of the side away from you. Place the socket on the circuit board, on the side facing you, in the position shown in the diagram below. It is essential that the socket be placed exactly as shown below. One corner of the socket will be marked with a notch or other mark to indicate pin #1. This corner should be at the top of the socket.

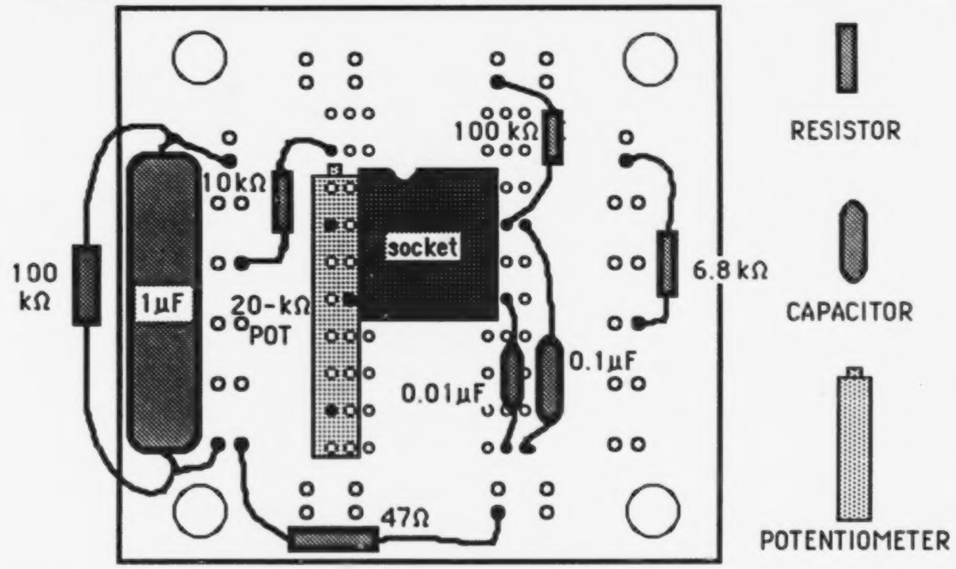


Solder all 8 of the pins to the plated side of the circuit board.

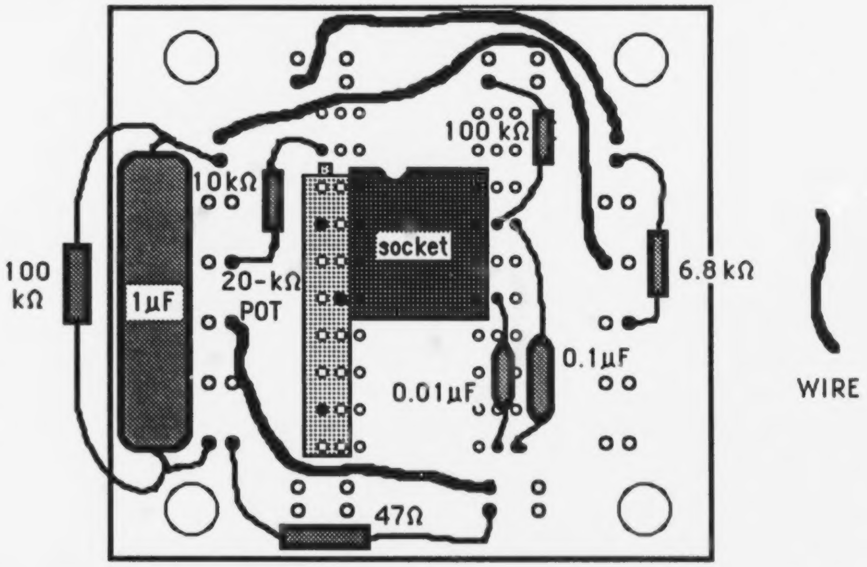
☐ 2. **Solder a 100-kΩ resistor and the 1-µF capacitor together.** Attach one of the 100-kΩ resistors in parallel with the 1-µF capacitor as shown below:



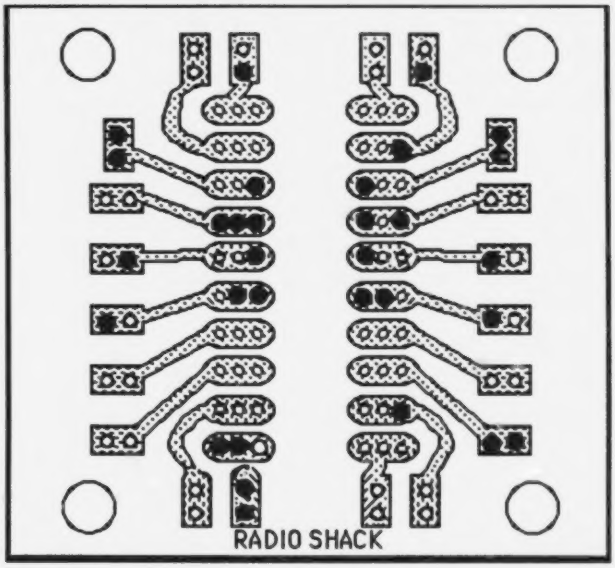
☐ 3. **Solder the resistors, capacitors and potentiometer to the circuit board.** Place all of the components except the LM331 chip into the board. Use the diagram below as a guide for positioning the components. Notice that the components go on the side without the copper plating. Carefully solder each of the leads to the plated side of the circuit board. Cut off the leads at the solder connections.


100 kΩ
10kΩ
100 kΩ
socket
1μF
20-kΩ POT
6.8 kΩ
0.1μF
0.01μF
47Ω
RESISTOR
CAPACITOR
POTENTIOMETER

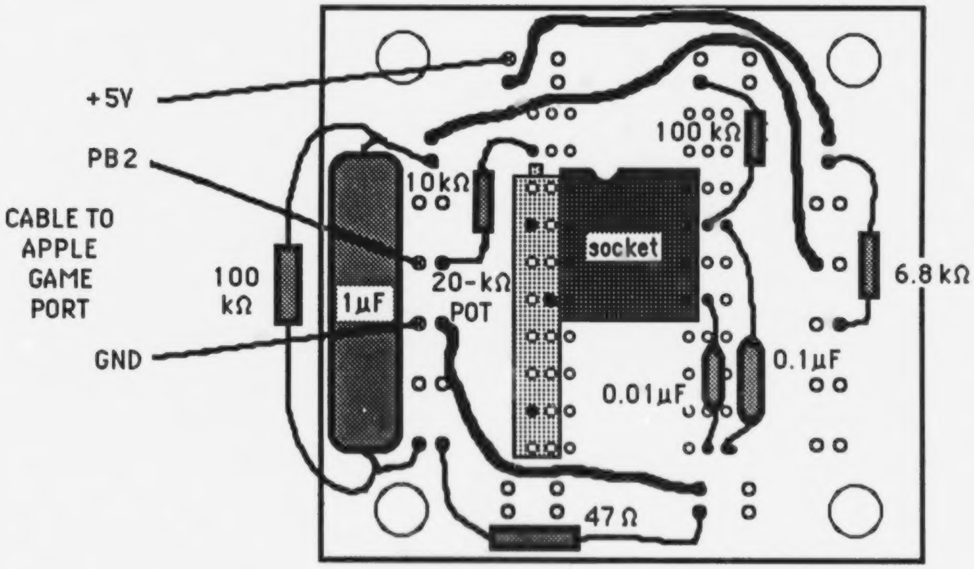
☐ 4. **Add the three jumper wires to the circuit.** Prepare the three short (approximately 5 cm long) wires. Strip a few millimeters of insulation off each of the wire ends. Add the three wires to the circuit board in the locations shown in the diagram.


100 kΩ
10kΩ
100 kΩ
socket
1μF
20-kΩ POT
6.8 kΩ
0.1μF
0.01μF
47Ω
WIRE

☐ 5. **Double-check your circuit**. Compare the back of your circuit board with the diagram of the back of the circuit board below. There should be an exact match of "filled holes" if you have everything right. Correct any mistakes you made in placing the components.



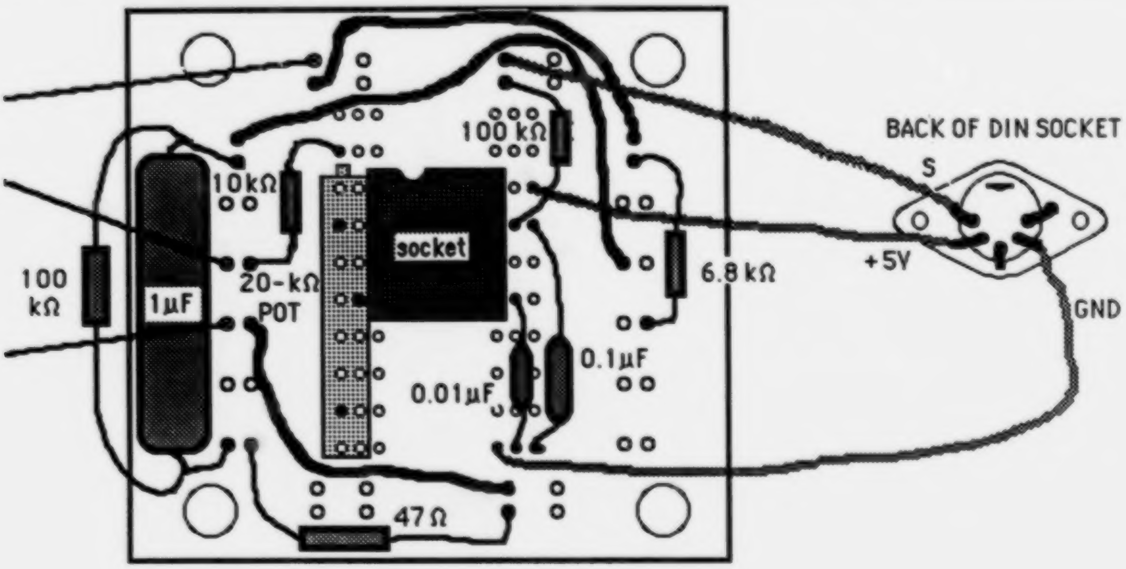
☐ 6. **Connect the leads from the game-port cable.** Refer to *Appendix A* and identify the following three leads: +5, PB2, and GND. Drill a hole in the side of the experimenter box for the cable to pass through. Thread the cable through the hole and tie a knot in the cable on the inside of the box. This knot serves as a "strain relief" to prevent the cable from being accidentally torn out of the box. Strip the insulation from the ends of the three cable leads and solder them to the locations shown below:



☐ 7. **Connect the DIN socket to the circuit board.** Make up three jumper wires (about 10 cm long) to extend from the circuit board to the DIN socket. Drill holes in the experimenter box for mounting the DIN socket. Solder these wires to the three pins on the DIN socket as shown on the diagram below. Thread the wires through the hole in the box and then connect the wires to the circuit board at the locations shown in the diagram. The illustration of the DIN socket shows the back of the socket with the holes facing away from you.[5]

---

[5] The choice of which terminals on the DIN socket to use for each wire is an arbitrary choice that we have made. The next three projects are designed with DIN plugs wired to match this wiring pattern.

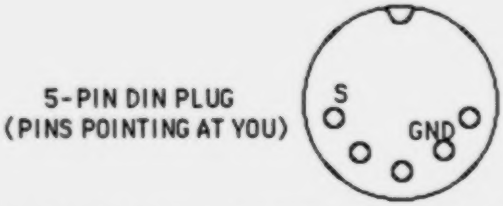


VOLTAGE MONITOR CIRCUIT BOARD

☐ 8. **Plug in the LM331.** The LM331 Precision Voltage-to-Frequency Converter can be damaged by static-electrical discharges. Allow any static charges to leak off of your body by touching a grounded metal object (such as a water pipe) before handling it. This is especially important in cold, dry weather. Insert the LM331 into the 8-pin IC socket. Pin #1 of the LM331 is marked with a dot. Be sure that the IC is inserted with pin #1 in the correct location (near the potentiometer).

☐ 9. **Prepare the test-probe cable.** For the cable, you need 2 conductors ending in a 5-pin DIN plug.[6] Two suggestions for this cable are: (1) Use a 5-pin DIN plug with two wires soldered onto it. Use the two terminals as shown in the diagram below. (2) Buy a 6-foot DIN patch cord that has a 5-pin DIN plug on each end and cut it in half. This gives you two ready-to-use cords. The extra cord can be used for another project.[7]

The two connections to use on the plug are shown below. If necessary, use an ohmmeter to identify the leads. The GND connection is connected to the ground of the power supply. The "S" lead will be used for the input signal. The voltage of this lead compared to ground will be measured by the Voltage Monitor.



Solder the test clips to the ends of the "S" and "GND" leads. Label the two test clips and plug the test-probe cable into the Voltage Monitor.

---

[6] Notice that only two of the three terminals that you connected on the DIN socket are used for this project. Other projects that use the Voltage Monitor will require the third (+5V) lead.

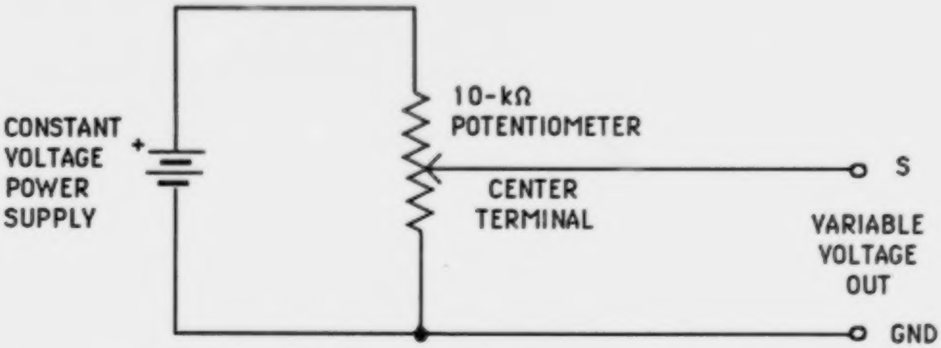
[7] Three other projects in this book that require a similar cord are *Thermocouple*, *pH Meter*, and *Strain Gage*.

☐ 10. **Test the completed circuit.** Turn off the computer and connect the Voltage Monitor to the game port. Place the *Project Programs* disk in the disk drive and turn on the computer. Run the VOLTAGE program. Choose the READ VOLTAGE option. Connect the test leads to a known voltage source. The voltage should be in the range 0 to 3.5 volts. A standard 1.5 volt battery is one possible source. Be sure to connect the S (signal) lead to the positive side of the voltage source. You may have to adjust the potentiometer to get things to work. If no voltage is registered, refer to the "Troubleshooting" section below.

☐ 11. **Calibrate the Voltage Monitor.** Connect a good quality voltmeter to the same voltage source that is connected to the Voltage Monitor. Adjust the potentiometer on the Voltage Monitor so that the computer registers 1 hertz per millivolt input (1000 Hz per volt). The potentiometer screw may need to be turned several complete rotations to reach the correct setting. Avoid touching the wires on the board as you turn the screw, as this may slightly affect the frequency reading.

☐ 12. **Mount the circuit in the experimenter box.** Mount the 5-conductor DIN socket in the hole in the side of the box. Place the circuit board inside the box and use screws or glue to fasten it inside. Attach the lid.

☐ 13. **Test the system over the voltage range.** Use a variable-voltage power supply for testing. If you do not have a commercial power supply with a voltage-output control, you can easily make one. Use a circuit such as the one below:



Connect the variable voltage to the Voltage Monitor and to a good voltmeter at the same time. Remember that the voltage used should not exceed 3.5 volts. For a number of different voltages between 0 and 3.5 volts, record the voltmeter and the program readings. If everything is working well, the readings should be very similar over the whole range. In case of problems, refer to the "Troubleshooting" section below.

Is the system accurate over its entire voltage range? What is the maximum error for the Voltage Monitor (assuming the meter you are using is perfect)? Are the voltage readings reproducible; that is, if you measure the same voltage several different times, do you get the same result? Is there any long term drift in the voltage readings?

Tips on Using the Voltage Monitor

- Do not use the Voltage Monitor with alternating-current (AC) electricity.
- The voltage input range on the Voltage Monitor is limited to approximately 0 to 3.5 volts. If the input voltage is outside this range, the display will read zero. This can cause confusion, especially when you watch a voltage reading gradually increase and then mysteriously go to zero (because it has exceeded 3.5 V).
- All voltage measurements made by the Voltage Monitor are with respect to the Apple's power supply ground lead. The GND test lead from the Voltage Monitor is a grounded wire.

- Be aware of polarity of the leads. The "S" test lead must always be more positive than the GND lead.

TROUBLESHOOTING

If the Voltage Monitor does not work properly, first double-check the following list of the most frequently made mistakes:

1. **Is the LM331 IC plugged in with pin #1 in the right place?** Pin #1 should be connected to pin #6 and to a 100-kΩ resistor. Are all eight pins seated properly?
2. **Are the leads from the Apple game port used correctly?** Make sure you are using the wires that are connected to the right pins. You should be using only the PB2, GND and +5V leads. Refer to *Appendix A* to determine which pin on the plug connects to each of these three leads. Use an ohmmeter to check for electrical conductivity between the proper pin and the circuit board.
3. **Are all of the components properly situated on the board?** Take a few minutes to make sure each lead goes in the proper hole. If possible, have someone else compare your circuit board with the diagrams included in the "Step-by-Step Instructions."
4. **Are all circuit board connections soldered properly?** Check the back (copper-plated) side of the circuit board. Have you soldered every lead sticking through a hole?
5. **Are there any "solder bridges" where there should not be?** Check the back of the circuit board. Look for places where solder has flowed in such a way as to connect two spots that should not be connected.
6. **Are the wires to the DIN socket and on the DIN plug used correctly?** Double-check the diagram in the "Step-by-Step Instructions." Also, the leads on the test probe are easy to confuse. Use an ohmmeter to make sure that the "S" test lead is really connected to the proper spot on the circuit board. Do the same with the GND test lead.
7. **Is the voltage you are trying to measure in the range 0 to 3.5 volts DC?**

If none of this helps, try checking some voltage levels on the circuit board with the Voltage Monitor connected to the computer and with the power on. Connect the negative lead of the voltmeter to pin #4 of the LM331. The table below indicates voltage levels that you should register as you attach the positive lead of the voltmeter to the other pins of the LM331.

PIN #	EXPECTED VOLTAGE	COMMENTS
8	+5 V	Power supply lead from game port. If you do not have +5 volts here, check your connections to the game port.
7	0 to 3.5 V	This is the input signal from voltage source Make sure that it changes as the voltage connected to the test probe changes.

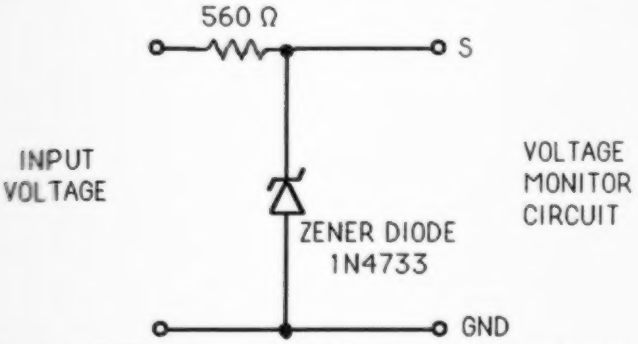
PROJECT EXTENSIONS

Circuit Modifications

The next three projects in this book are circuits you can build to use with the Voltage Monitor. The circuits allow you to measure temperature, pH and force. Several modifications of the Voltage Monitor circuit itself are suggested below.

◊ Zener-diode input protection

The Voltage Monitor can read voltages only up to +3.5 volts. If a slightly higher voltage (up to 5 volts or so) is connected across the test leads, a zero voltage will be registered, but the Voltage Monitor will not be damaged. If a very large positive voltage is connected across the test leads, the LM331 IC could be damaged. To prevent this, a zener diode and a resistor can be added to the input leads of the Voltage Monitor. The connection of these two new components is shown below:

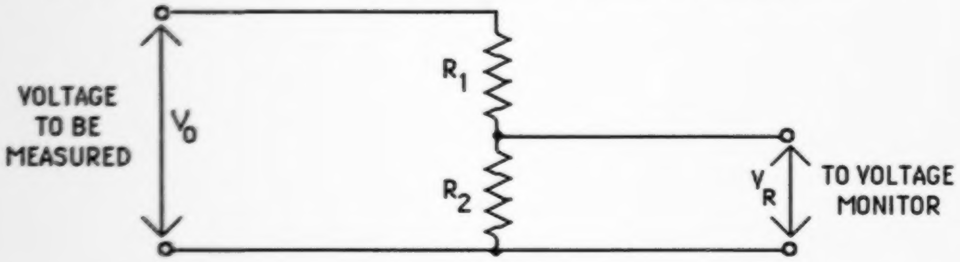


If the voltage applied to the "S" line exceeds 5.1 volts, the zener diode begins conducting electricity. The voltage applied to pin #7 of the LM331 should never exceed 5.1 volts with this addition.

The zener diode will also protect the Voltage Monitor from any negative voltage inputs. If the signal-test lead is connected to a negative voltage, the zener diode conducts as a normal diode would and pin #7 of the LM331 stays at the GND voltage level.[8]

◊ Voltage dividers

If the voltage that you want to measure is higher than 3.5 volts, you can use a simple "voltage divider" to expand the voltage range. A voltage-divider circuit is shown below:



The voltage that is delivered to the Voltage Monitor is:[9]

$$V_R = V_0 \times R_2 / (R_1 + R_2)$$

Notice that V_R is always smaller than V_0. The ratio of $R_2 / (R_1 + R_2)$ determines the division factor. We recommend using 10 kΩ for R_2 and selecting R_1 to get the proper division factor.[10] Do not attempt to use this method to study voltages greater than approximately 20 volts; larger voltages can be harmful to people and circuits.

◊ Amplification

If the voltages you wish to monitor are smaller than a few tenths of a volt, you may want to amplify the signal before it gets to the Voltage Monitor. This allows for much more precise measurement of small voltages. A simple linear amplifier that can be used for this purpose is shown below. The heart of this circuit is an operational amplifier IC, better know as an *op amp*. These devices are very inexpensive (about a dollar) and work remarkably well.[11] An op-amp circuit can amplify a voltage signal to many times its original value. The ratio of the output voltage to the input voltage for a circuit is called the *gain*. For the circuit shown below, the gain is equal to:

---

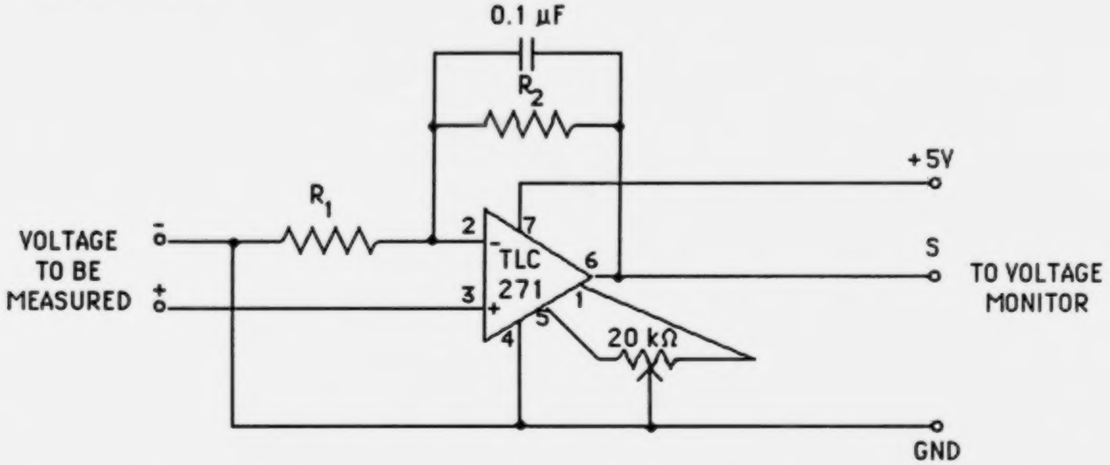
[8] For information on zener diodes, see *The Art of Electronics* by Horowitz and Hill, pp.11-12 or *PSSC Physics*, 5th Edition by Uri Haber-Schaim, et al, D.C. Heath and Co., Lexington, MA (1981) pp. 335-336.

[9] See *The Art of Electronics* by Horowitz and Hill, p. 7.

[10] If the resistors you choose are too small, you may draw so much current through the voltage divider that you affect the voltage reading. If the resistors you choose have values that are too high, the voltage division will not come out exactly as predicted by the equation shown above. This is due to the small current that flows into the Voltage Monitor.

$$\text{gain} = 1 + R_2 / R_1$$

Notice that the gain can never be less than one. You can select the gain you want by using the proper values for the resistors R_1 and R_2.[12] When using this amplifier with a Voltage Monitor, you should set the gain so that the voltage delivered to the Voltage Monitor will vary from about 0 volts to about 3 volts.[13]



The op amp in this circuit is powered by the +5V line from the Apple game port. A *single-supply op amp* must be used. Many op amps require both negative- and positive-power supply leads. A single-supply op amp can operate using the +5V and GND leads for power. This circuit is designed to use the TLC271 CMOS op amp. The input voltage delivered to pin #3 of the op amp must be positive with respect to GND. The output is proportional to the input with the gain = $1 + R_2 / R_1$. The potentiometer connected to pins #1 and #5 allows you to set the *offset voltage*. Adjust it to make sure that the voltage at pin #6 is zero or slightly positive when the input voltage is zero.[14]

Program Improvements

The VOLTAGE program on the disk is a fairly simple program that can be improved by adding new features. The "Program Design Notes" section below explains how the VOLTAGE program operates. It should help you in making modifications. Some suggested changes are given below:

◊ Modify the program to print the data table on your printer.
◊ Allow the voltage and time data to be saved on a disk and reloaded later.

---

[11] If you have never worked with operational amplifiers, you are encouraged to read about these interesting devices in one of the electronics reference books listed in *Appendix C*.

[12] You can even use a potentiometer for one of these resistors and have an amplifier with a gain which is adjustable.

[13] R_1 and R_2 should be in the range 1 kΩ to 1 MΩ. Don't expect to get a gain of more than about 60 without problems. If more gain is needed, use several amplification stages. Project 10 (*Strain Gage*) uses a multistage amplifier.

[14] If a different op amp is used, the offset circuit (pins #1 and #5) may have to be changed.

◊ If your computer has a "clock card" to keep track of real time, change the program so that it reads the clock time and stores it along with the voltage reading.
◊ Add a mode to the program that graphs the voltage readings versus time.
◊ Add a mode to the program that displays the voltage in large low-resolution graphics block digits so they can be read at a distance.
◊ If you are using either a voltage divider or an operational amplifier with your Voltage Monitor (both are mentioned as possible circuit modifications), change the VOLTAGE program so that it reads the correct voltage when these extra circuits are used.

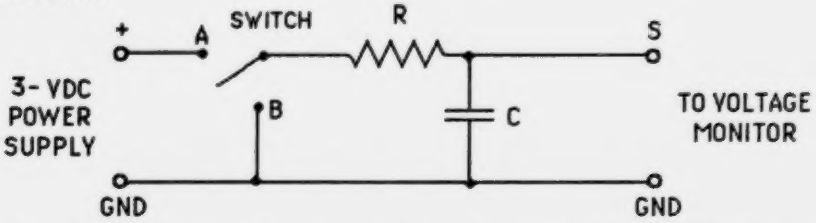
Additional Experiments

◊ A computerized study of batteries

Set up a circuit that consists of a small lamp powered by a AA cell. Attach the test leads from the Voltage Monitor across the AA cell. Use the VOLTAGE program (mode S) to check the voltage across the battery every 10 minutes or so. When the AA cell is completely dead, take a look at the table of voltage readings to get an idea of how the battery held up. Graph the voltage across the battery as a function of time. Repeat this experiment with other types and brands of cells and compare results. The experiment can also be repeated with different electrical loads.

◊ Charge and discharge curves for capacitors

The Voltage Monitor can be used to study the charging and discharging of a capacitor. Hook up the circuit shown below:



To study the charging of the capacitor, move the switch to position A. To discharge the capacitor, move the switch to position B. Since the Voltage Monitor reads the voltage only about once a second, you will want the capacitor to charge and discharge slowly. Use very large resistors and capacitors. The following values work nicely: R = 1 Megaohm and C = 4.7 microfarad

SUPPLEMENTAL INFORMATION

Program Design Notes

The VOLTAGE program on the disk will do the calculations necessary to convert the frequency readings from the Voltage Monitor to voltage readings. Here is a section-by-section explanation of the program:

Lines 100 to 990: Initialization
This section loads the machine-language FREQ program.[15] It also displays the main menu.

Lines 1000 to 2990: Read Voltage
This section simply displays the count and voltage readings continuously.

---

[15] The FREQ program assumes that the signal is coming into the computer on the PB2 game-port lead. It is possible to make a modification so that the input signal can be received on the PB0 line. This modification is necessary to use the VOLTAGE program with a Vernier Software 9-pin Voltage Input Unit. Refer to the Teacher's Guide (*Appendix G*) for further information.

Lines 2000 to 2990: Read Voltage and Save in Memory
This section displays the voltage readings and also saves them in memory. When this mode is selected, you will be asked:

```
ENTER DELAY TIME (TIME BETWEEN VOLTAGE
READINGS STORED IN MEMORY) [1]
```

This option allows you to save voltage measurements made over a long time period in memory. This is often useful in many experiments. If you enter a time delay of 5, for example, the voltage reading will be checked every second, but the readings will be saved in memory only every 5 seconds. In most cases, you do not need this delay. If you simply press <RETURN> or enter a 1, the readings will be saved in memory as fast as they are made, which is about once a second. A maximum of 500 voltage readings can be saved.

After you enter this value, the program will begin monitoring the voltage. Any time the voltage is saved in memory, the line on the data table is marked "<- SAVED". The screen will display both the time (in seconds) and the voltage.

Lines 4000 to 4990: Table of Data
This section lets you look over a table of data collected using the S = SAVE VOLTAGE DATA IN MEMORY option. Both the voltage and the time at which it was measured are displayed.

Lines 10000 to 10990: Voltage Check Subroutine
These lines do the actual checking of the voltage from the Voltage Monitor. Here is a line-by-line explanation of how that is done:

10020 CALL 33536 : This BASIC statement transfers control to the machine-language program FREQ. The FREQ program counts the pulses coming from the Voltage Monitor for 0.501563 seconds. The result is stored in memory locations 250 and 251 ($FA and $FB in hexadecimal notation).

10030 C = PEEK (250) + 256 * PEEK (251) : After the machine-language routine is finished counting, control return to BASIC and this statement is used to read the count. Each unit in location 251 represents 256 counts.

10050 F = INT (C / 0.501563 + .5) : The frequency is calculated by taking the counts registered and dividing by the time period over which the counting took place. Using the FREQ program, that time interval is always 0.501563 seconds.

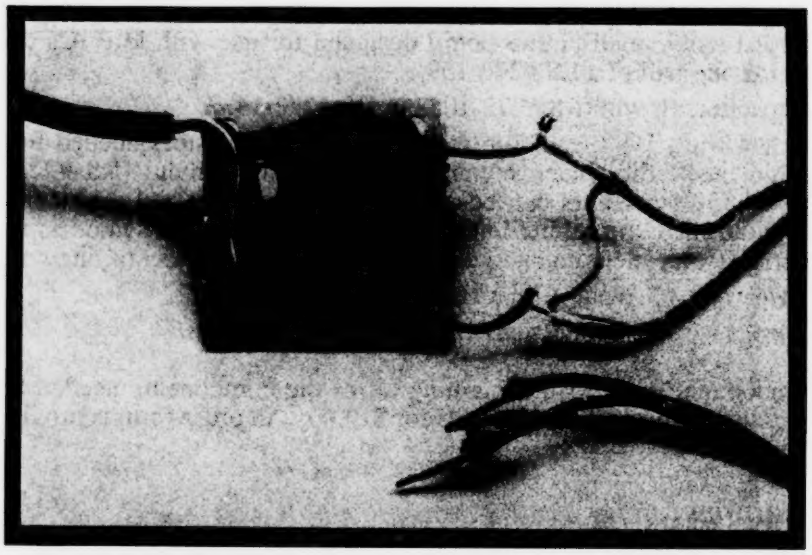
10060 V = F/1000 : V = INT (V * 100 +.5)/100 : The program assumes that your Voltage Monitor is set for 1000 hertz output per volt. This step of the program makes this conversion. It then rounds the result to the hundredth of a volt.

10090 RETURN : This statement ends the voltage reading subroutine.

The following variables are used in the VOLTAGE program:

C	Counts from Voltage Monitor
D$	Control D, used for disk access
F	Frequency
I	Delay loop counter
J	Data line counter
M$	Mode of operation
N	Number of data lines
Q$	Dummy input string
T	Time (in seconds)
TD	Time delay for saving data
T( )	Time data saved in memory
V	Present voltage reading
V( )	Voltage data saved in memory

PROJECT 8: THERMOCOUPLE



In this project you will build a temperature-measurement system that will allow you to measure temperatures up to 1300 degrees Celsius. The temperature sensor used is a *thermocouple*. A thermocouple consists of two sections of wire made from different metals. The thermocouple produces a small voltage that varies with the temperature. A simple amplifier circuit is used to increase this voltage. The voltage can then be fed into the Apple using the Voltage Monitor from Project 7. This temperature-measurement system is not as accurate as the IC Temperature Probe (Project 6), but it covers a much wider temperature range.

PROJECT REQUIREMENTS

Project Prerequisite

The Voltage Monitor from Project 7 is needed for this project.

Background Knowledge

- Electronics: Experience with soldering is recommended. Specific component placement on the circuit board is not given for this project, so you have to be able to read a simple schematic diagram and decide how to position the components.
- Programming: A simple BASIC program change may be required for calibrating the system. Some of the project extensions require experience with BASIC programming.

Parts

- ☐ type K (Chromel/Alumel) thermocouple wire, 50 cm or longer (Catalog #GG-K-24 from Omega Engineering, One Omega Drive, P.O. Box 4047, Stamford, CT 06907, (203) 359-1660. It is sold in 25-foot minimum lengths.)
- ☐ TLC271 op amp (This is a low-power CMOS op amp made by Texas Instruments. It is specifically designed to be used with a single +5-volt power supply. It may be available as Radio Shack #276-1748. It was included in their 1986 catalog.)
- ☐ 8-pin IC socket (RS #276-1995)

- ☐ 20-kΩ, multiturn potentiometer (RS #271-340)
- ☐ cable with 5-pin DIN connector (one-half of a DIN patch cord [RS #42-2151] or a 5-pin DIN plug [RS #274-003] with 3 wires soldered onto it)
- ☐ circuit board (Any small circuit board designed for use with DIP ICs can be used. One possibility is one-half of a RS #276-159.)
- ☐ 0.1-μF capacitor, 10 volt (RS #272-1053 or RS #272-1432)
- ☐ 100-kΩ, 1/4-watt, 1% metal-film resistor (The two resistors needed for this project are included in the Radio Shack metal-film resistor assortment [RS #271-309]. Carbon-composition [5%] resistors can be used, but since their exact resistance value is not as precisely controlled, the calibration of the system may not be as good as it is when you use metal-film resistors. See *Appendix B* for the color codes for identifying resistors.)
- ☐ 2.2-kΩ, 1/4-watt, 1% metal-film resistor
- ☐ miscellaneous wire for connections on the circuit board

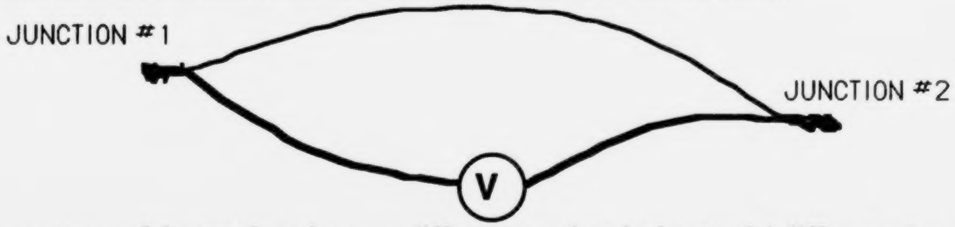
Note: A "Thermocouple Parts Kit" containing all of the components needed for this project is available from Vernier Software. This kit sells for $13.00. An order form is provided at the back of this book.

Tools and Supplies

Soldering iron, solder, wire cutters, wire strippers and two pairs of pliers.

BACKGROUND INFORMATION

A thermocouple is a simple, inexpensive method of measuring temperatures. To make a thermocouple, all you need is two sections of wire made of different metals. Join the ends of the two wires together by welding, soldering, or just twisting them together to form a loop. Now cut one of the two wires in half and connect the two ends to a voltmeter as shown below:

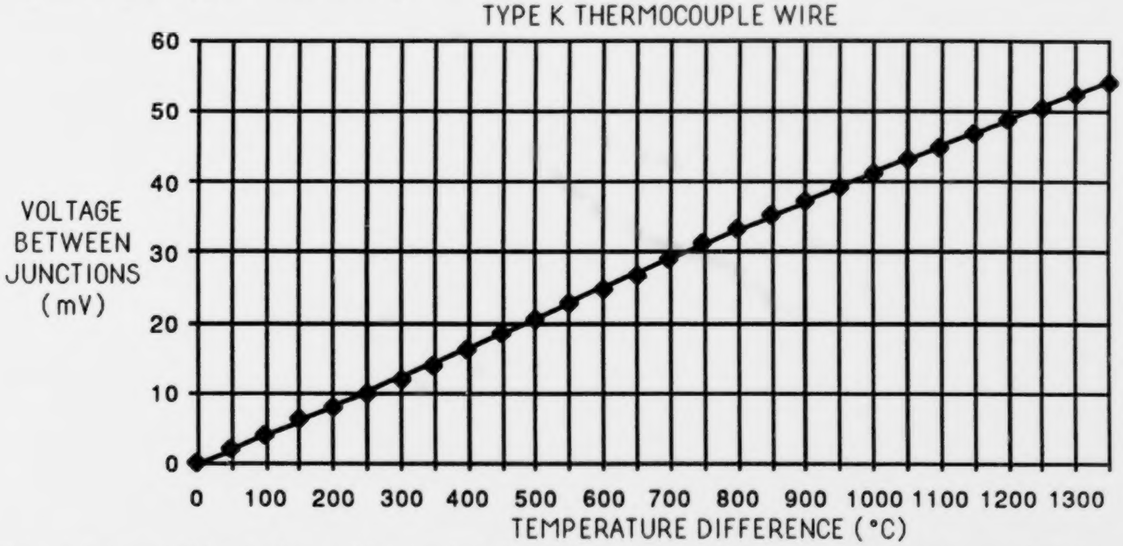


If the temperatures of the two junctions are different, an electrical potential difference is created and the voltmeter will read a small voltage. This production of a voltage from the temperature difference at the two junctions is called the *Seebeck Effect*. As the difference in temperatures between the two junctions changes, the voltage changes. In fact, the voltage produced is very nearly proportional to the difference in the temperatures of the two junctions. This makes the Seebeck Effect voltage from a thermocouple an effective method of measuring temperature differences.[1]

In theory, any two different metals could be used to make up the wires of the thermocouple. For most applications, however, commercial "thermocouple wire" is used. Pairs of wires of different alloys, selected for their usefulness as a thermocouple, are sold together. The thermocouple wire is identified by its *type*. The type is a letter designation that identifies the two different metals used in the wires. Common thermocouple types are T, E, K S, R, B, and J. For this project, we recommend type K thermocouple wire. Type K thermocouple wire is made of the two alloys, Chromel and Alumel. One of the advantages of using a commercial thermocouple wire is that the Seebeck Effect response of the wire pair is known. The graph below is taken from data provided by Omega Engineering for type K thermocouple wire.[2]

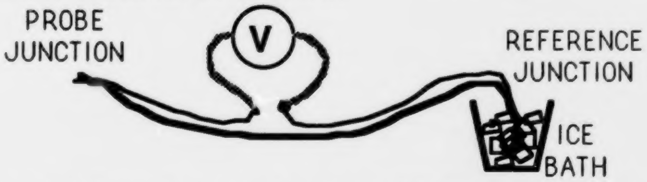
---

[1] Refer to "Thermocouple" in *Van Nostrand's Scientific Encyclopedia* or a college physics text, for example, *College Physics*, 4th Edition by Sears, Zemansky and Young, Addison-Wesley, Reading, MA, 1974, pp. 448-451.



Notice several things about this graph: (1) The voltage produced is nicely proportional to the temperature difference between the junctions; (2) The thermocouple can be used with large temperature differences; and (3) The electrical potential differences (voltages) produced by the thermocouple are never very large (at most, about 60 mV or 0.06 V).

The simplest way to build a thermocouple temperature-measurement system is to hook up the type K thermocouple wire with a voltmeter as shown below:

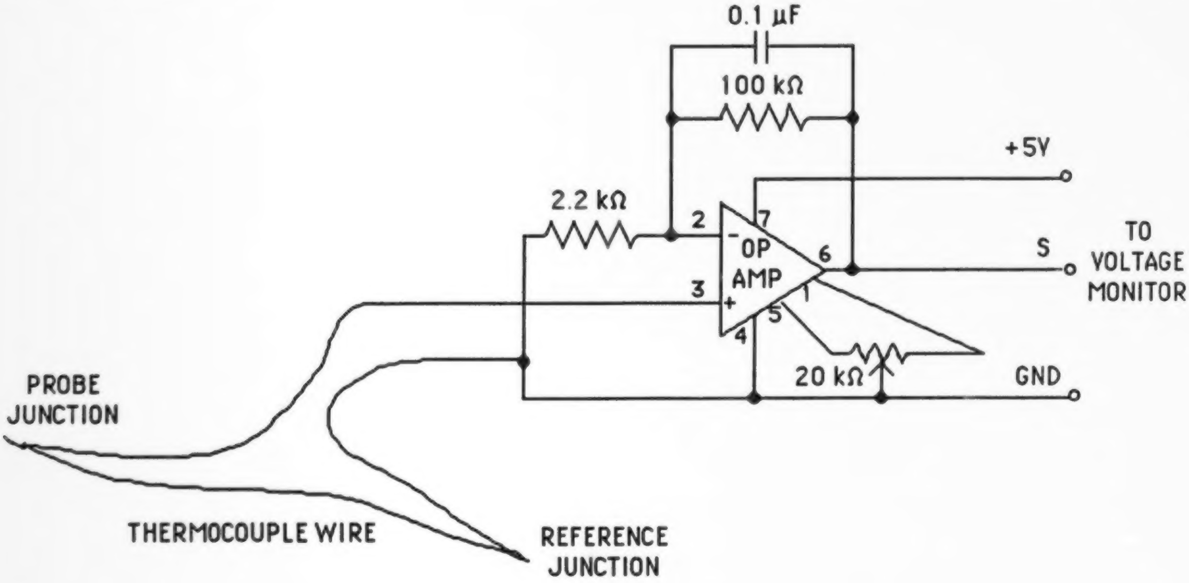


In most thermocouple applications, the two wires are welded or soldered together using silver at the junctions. It is possible to get good results by just twisting the two wires together very tightly. One of the junctions (referred to as the *reference junction*) is kept in an ice water solution so that it is at approximately 0°C. The other junction is used as a temperature probe. A good quality (high-impedance) voltmeter is required. Most digital voltmeters that are able to read millivolts will work. The voltage can be read from the meter and then the graph above can be used to determine the temperature difference between the junctions. If the reference junction is at 0°C, the temperature difference is the temperature of the probe in degrees Celsius.

We will use a very similar system to read temperatures from the thermocouple into the computer. We will need to amplify the voltage produced by the thermocouple so that it is in the range that can be read accurately by the computer. This requires a simple operational amplifier circuit. The recommended circuit is shown below:

---

[2] Data on the response of all types of thermocouple wire are listed in *Temperature Measurement Handbook and Encyclopedia* from Omega Engineering, One Omega Drive, P.O. Box 4047, Stamford, CT 06907, (203) 359-1660.



SCHEMATIC DIAGRAM OF THERMOCOUPLE AMPLIFER

The output of this amplifier is a voltage signal that varies 46.5 times more than the unamplified voltage signal from the thermocouple. This signal can be fed directly into the Voltage Monitor that was constructed in Project 7. The 20-kΩ potentiometer is used to adjust the *offset* of the circuit's voltage output. The offset is the voltage the circuit produces when the two thermocouple junctions are at the same temperature. A program on the *Project Programs* disk will read the voltage from the Voltage Monitor and convert it to a temperature. The conversion from voltage to temperature is done using the proportionality of voltage and temperature shown on the type K thermocouple graph and the known gain of the amplifier.[3]

The thermocouple can be used over a wide range of temperatures. It can be used to show the difference in temperature of different flames or to check the temperature in an oven.

An excellent free source of information on thermocouples is the *Temperature Measurement Handbook and Encyclopedia* from Omega Engineering. This catalog of temperature measurement equipment also includes useful information on calibrating and using thermocouples.

STEP-BY-STEP INSTRUCTIONS

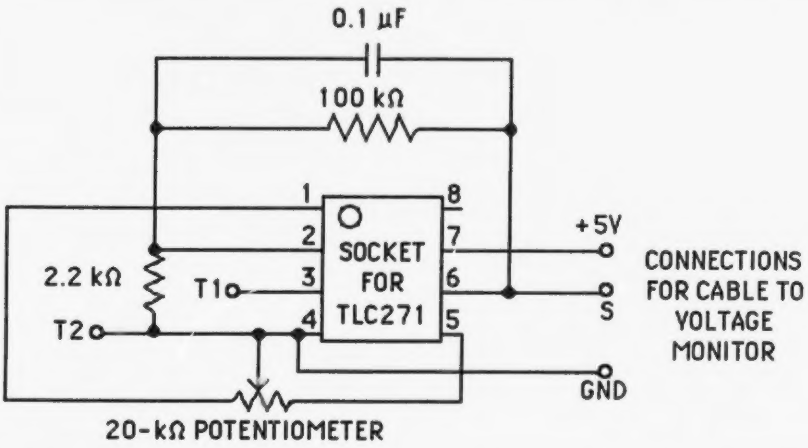
☐ 1. **Solder the 8-pin socket on the circuit board.** Place the socket near the center of the unplated side of the circuit board. One of the corners of the socket will be marked with a notch or other mark to indicate where the #1 pin of the IC should be placed. The other holes of the socket are referred to by number. They are numbered in a counterclockwise pattern as you view the socket from the top (with the pins of the socket pointing away from you). Solder all 8 of the pins to the plated side of the circuit board.

☐ 2. **Assemble the amplifier circuit.** Solder the two resistors, one capacitor, and one potentiometer to the leads of the socket. The working diagram below shows the proper positioning of the components. Use short jumper wires to make connections as you need them. Note carefully the wiring of the potentiometer. All three terminals are used. The two outside terminals connect to pins

---

[3] The details of this calculation are explained in the "Supplemental Information" section below.

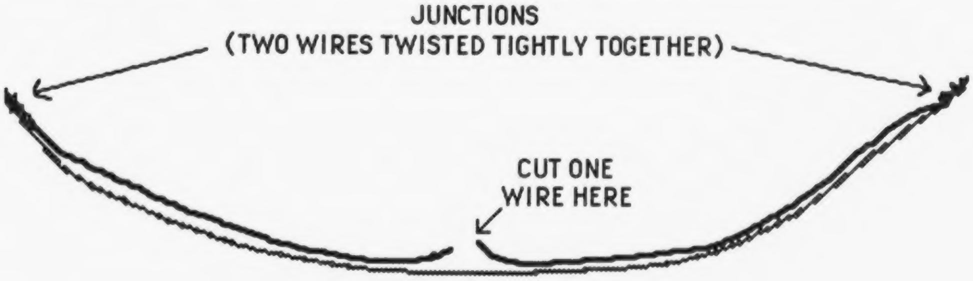
#1 and #5 of the IC. The middle terminal is connected to pin #4 of the socket. You will probably need to use jumper wires for some of these connections. Leave holes at the two locations labeled T1 and T2 for the thermocouple connections. Leave holes for the three leads of the cable to the Voltage Monitor.



WORKING DIAGRAM OF THE THERMOCOUPLE AMPLIFIER CIRCUIT (TOP VIEW)

☐ 3. **Prepare the thermocouple wire.** Cut a section of type K thermocouple wire at least 50 cm long. Strip two centimeters of insulation from both wires on one end. With two pairs of pliers, twist the bare ends of the two wires **tightly** together. Make at least a dozen or so twists. Repeat this procedure with the other end of the wire.

Now cut the middle of **one** of the wires of the thermocouple pair as shown below:



THERMOCOUPLE WIRE PAIR

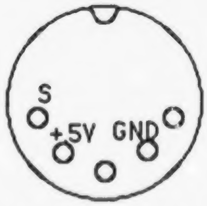
The cut ends are to be connections to the amplifier circuit. Solder them to the locations marked T1 and T2 on the amplifier circuit. These are thick wires; you will have to heat them a while to make a good solder connection.

☐ 4. **Prepare the cable to the Voltage Monitor.** Make up a 3-conductor cable to extend from the amplifier circuit to the Voltage Monitor (Project 7). If you followed our suggestion in constructing the Voltage Monitor, you will have a 5-pin DIN socket on the box for this connection. For the cable, you need 3 conductors ending in a 5-pin DIN plug. Two suggestions for this wire are: (1) Use a 5-pin DIN plug with three wires soldered to it; or (2) Buy a 6-foot DIN patch cord that has a 5-pin DIN plug on each end and cut it in half. This gives you two ready-to-use cords. The extra cord can be used for another project.[4] The three connections to use on the cord are shown below. If necessary, use an ohmmeter to identify which wire on your cable connects to which terminal on the plug.

---

[4] Two other projects that require a similar cord are *Strain Gage* (Project 10) and *pH Meter* (Project 9).

5-PIN DIN PLUG
(PINS POINTING AT YOU)



Solder the three connections to the locations marked "+5V", "S" and "GND" on the working diagram.

☐ 5. **Plug in the op amp.** Carefully place the integrated circuit into the socket. Be sure to orient pin #1 at the proper corner of the socket.

☐ 6. **Adjust the offset.** Turn off the computer. Plug the thermocouple circuit into the Voltage Monitor. Connect the Voltage Monitor to the Apple game port. Turn on the computer and run the THERMOCOUPLE program on the *Project Programs* disk. With the program running and both junctions of the thermocouple at room temperature, adjust the offset potentiometer until the temperature difference displayed is zero. You may need to make several turns of the screw on the potentiometer. If you cannot get the reading adjusted to zero, refer to the "Troubleshooting" section below. Once the offset is set, it should not need to be adjusted again.

☐ 7. **Test the completed circuit.** Using a match, try heating each of the thermocouple junctions as you watch the temperature display. Heating one junction should cause the temperature reading to rise. Label this end "probe." Label the other end "reference." If the temperature reading does not respond when either end is heated, refer to the "Troubleshooting" section.

☐ 8. **Use the system to measure various temperatures.** Place the "reference" junction in an ice/water solution. The temperature reading displayed will be the difference between the temperatures of the two junctions. Since the reference junction is at 0°C, the reading will be the Celsius temperature of the probe junction. Try checking the calibration of your system by measuring the temperature water at several temperatures up to the boiling point (100°C). If the temperature displayed is off by more than 10°C, you might want to adjust the calibration. This is explained in the "Troubleshooting" section.

How accurate is this temperature-measurement system? If possible, check the accuracy of the system at higher temperatures.[5] Are the results reproducible? This system assumes that the response of the thermocouple is proportional to temperature. Is it? Is there any long-term drift of the calibration of the system?

Try measuring the temperature of various flames. The type K thermocouple wire can be used up to 1350°C, high enough to study most common flames. Are candle flames as hot as matches? How hot are disposable-lighter flames? Where are the hottest spots in flames? How is the color of a flame related to the temperature?

TROUBLESHOOTING

If the thermocouple circuit is not working properly, check for these most common errors:

◊ Is your op amp inserted properly? Pin #1 is marked with a dot or notch. Is it where it should be? Are all the pins seated properly?

---

[5] If you have an oven that is is accurately controlled, you could check the calibration with it. One other possibility is to measure the melting point of copper (1084°C) or aluminum (660°C) with the thermocouple.

◊ Have you made the connections from the Voltage Monitor correctly?
◊ Does your TLC271 get hot when you plug in the circuit? Touch the top of the op amp. If it is hot, you must have a serious wiring problem, and you may even need to replace the IC after you fix the wiring error.
◊ If your readings are fluctuating widely, you probably have a bad connection between the thermocouple wire and the circuit board.
◊ Are the thermocouple wires tightly twisted together (at least a dozen times) at the junctions?
◊ Visually double-check all of the connections on the op amp. In particular, do you have the 20-kΩ potentiometer wired correctly?
◊ Make sure your Voltage Monitor is working properly. Connect it to a known voltage source and test it.

If none of these problems are found, it is time to do some troubleshooting using a voltmeter. Connect the thermocouple circuit to the Voltage Monitor. Plug the Voltage Monitor into the game port and turn on the computer. Set the voltmeter to read DC voltages between 0 and 5 volts and connect the negative lead of the voltmeter to pin #4 of the TLC271 IC. Check the voltage at the following pins of the TLC271:

PIN #	EXPECTED VOLTAGE	COMMENTS
7	+5 V	Power supply lead from Voltage Monitor. If you do not have +5 volts here, check your connections to the Voltage Monitor.
3	0 to 60 mV	This is the input voltage from the thermocouple. It should be 0 volts if the two junctions are at the same temperature. If the probe junction is heated with a match, a few millivolts should be read.[6] If the voltage reading is negative when the junction is heated, you are using the wrong junction as the probe. If no voltage is registered when the probe junction is heated, you must have a problem with your thermocouple. Check to be sure the two wires are tightly twisted together. Also, make sure that you have a good connection between the thermocouple wire and the circuit board.
6	+0.05 to +3 V	This is the output voltage of the amplifier. It should increase by a volt or so as the probe junction is heated with a match. This voltage should also be affected by adjustment of the 20-kΩ potentiometer. If the potentiometer has been set properly, this voltage should be 0.05 volts when the two thermocouple junctions are at the same temperature. If you do get a voltage that varies as it should here, you must have a problem with the connections to the Voltage Monitor or with the Voltage Monitor itself.

Adjusting the calibration of the system:

If you use the recommended resistors and thermocouple wire in this project, the calibration of the system should be fairly accurate. If it is not, you can make a change to the BASIC program to correct for calibration errors. Line 224 of the program THERMOCOUPLE should be changed. Substitute a larger or smaller number in place of the 46.5 in this line. A larger number will cause the temperature readings to increase; a smaller number will decrease the displayed temperature.

---

[6] This test may not work with an inexpensive, analog meter.

PROJECT EXTENSIONS

Circuit Modifications

- Solder flexible lead wires between the thermocouple and the amplifier. This allows the thermocouple to be moved around easily. For best results, use *coaxial cable* for this extension. The coaxial cable needed is a single conductor inside a metal shield. Connect the shield to pin #4 of the TLC271 and then to one of the thermocouple leads. Connect the inside conductor to pin #3 of the IC and then to the other thermocouple lead.[7]

- Place the thermocouple amplifier circuit in an experimenter box. Select a box large enough for the circuit board. Drill holes in the box for the thermocouple wire and the cord to the Voltage Monitor.

- Weld or silver-solder the thermocouple junctions. In most industrial uses of thermocouples, the junctions are welded together permanently. This will make the junction more durable. You can even use ordinary solder to connect the junctions if you are willing to limit your temperature measurements to below the melting point of solder. The calibration of the thermocouple is not affected by the welding or soldering.

- Use the thermocouple to measure low temperatures. Type K wire can be used to measure to –270°C. Use the "reference" junction as the "probe" and vice versa.

- Try other types of thermocouple wire. Other types of thermocouple wire can be used with the same amplifier circuit. Some types have higher temperature ranges; some can be used at very low temperatures. If you change to a different type of thermocouple wire, you will need to change the statements in the THERMOCOUPLE program that convert the voltage to a temperature.

- Build a cold-junction compensation circuit. It is possible to use an electronic circuit to take the place of the ice/water bath for the reference junction of the thermocouple. A small circuit, which includes an AD590 temperature transducer, "simulates" a reference junction in an ice bath. One reference on cold-junction compensation circuit is *Transducer Interfacing Handbook* by Daniel H. Sheingold.

Program Improvements

The THERMOCOUPLE program on the *Project Programs* disk is fairly simple. It can be improved by adding new features. The "Program Design Notes" below explain how the THERMOCOUPLE program operates. It should help you in making modifications. Some suggested changes are given below:

- Change the program so that it can display temperatures in Fahrenheit and kelvin degrees.
- Add an option to display or print a table of temperature versus time.
- Allow the temperature and time data to be saved on a disk and reloaded later.
- If your computer has a "clock card" to keep track of real time, change the program so that it reads the clock time and stores it along with the temperature.
- Program the computer to record the maximum and minimum temperatures.
- Add a mode to the program to graph the temperature readings versus time.
- Add a mode to the program that displays the temperature in large block digits so they can be read at a distance.

---

[7] These solder joints will form two new, dissimilar metal junctions that could also produce a potential difference due to the Seebeck Effect. They are like a second thermocouple. This second thermocouple will not cause any problems since its two junctions are kept at the same temperature.

Additional Experiments

The completed thermocouple temperature-measurement system can be used for many interesting projects. A few examples are listed below:

◊ Check the calibration of ovens.
◊ Experimentally determine the melting point of copper, bismuth, or other solids.
◊ Study how the temperature inside a bunsen burner flame varies at different heights above the barrel of the burner.
◊ Collect and graph data on how a hot object cools. How is the temperature change mathematically related to the time during which the object has cooled?

SUPPLEMENTAL INFORMATION

Program Design Notes

The THERMOCOUPLE program on the *Project Programs* disk will do the calculations necessary to convert the voltage readings from the thermocouple circuit into temperatures. Here is how it makes that conversion.

The graph of thermocouple voltage versus temperature for type K thermocouple wire from the "Background Information" section above is approximately a straight line. This shows that the voltage produced is proportional to the temperature difference. Using the voltage produced at 1000°C (0.041 V) to determine the proportionality constant, the equation relating temperature to voltage is:

$$\begin{aligned}\text{temperature} &= (1000°\text{C} / 0.041 \text{ volt}) \times \text{thermocouple voltage}\\ &= (24{,}400°\text{C} / \text{volt}) \times \text{thermocouple voltage}\end{aligned}$$

The voltage measured by the Voltage Monitor is the thermocouple voltage amplified by the op amp circuit. The voltage is multiplied by a *gain* factor. For this type of circuit (called a noninverting amplifier), the gain is determined by the ratio of the resistors used. The equation used is:

$$\text{gain} = 1 + R_2 / R_1$$

where: R_1 is the resistance between pins #2 and #4 of the op amp

and: R_2 is the resistance between pins #2 and #6 of the op amp[8]

The values used in this project are $R_1 = 2.2$ kΩ and $R_2 = 100$ kΩ. This yields a gain for our amplifier of:

$$\text{gain} = 1 + R_2 / R_1 = 1 + 100 \text{ k}\Omega / 2.2 \text{ k}\Omega = 46.5$$

The reason that precision (1%) resistors were recommended for this project was so that we would know this amplification factor fairly accurately.

The voltage read by the Voltage Monitor is 46.5 times the thermocouple voltage. If the op amp produced a zero output voltage when both thermocouple junctions were at the same temperature, we could just divide the output voltage by 46.5 to determine the thermocouple voltage. The *offset* of the operational amplifier adds a slight complication. The offset is the output voltage produced by the amplifier when there is no input signal (when both thermocouples are at the same temperature). The 20-kΩ potentiometer connected to pins #1 and #5 of the TLC271 allows you to control the offset. We could set the offset to zero, but it is better to set the offset to a small positive voltage in this case. The reason for this is that the Voltage Monitor cannot read negative voltages. As you adjust the 20-kΩ potentiometer to set the offset, the Voltage Monitor will report the voltage as "0.000" any time the voltage is zero or negative. Because of this, it is easier to set the offset at a small positive voltage. The THERMOCOUPLE program assumes that the offset is set at 0.05 volts.[9]

---

[8] For a derivation of this equation and more information on op amp circuits, refer to an electronics text such as *The Art of Electronics* by Horowitz and Hill.

The voltage read by the Voltage Monitor is:

Voltage Monitor voltage = thermocouple voltage x 46.5 + 0.05 volts

thermocouple voltage = (Voltage Monitor voltage - 0.05 volts) / 46.5

Combining this with the equation connecting thermocouple voltage and temperature yields:

temperature = thermocouple voltage x 24,400°C / volt

= (Voltage Monitor voltage - 0.05) / 46.5 x 24,400°C / volt

This is the equation used by the THERMOCOUPLE program to calculate temperatures. The voltage from the Voltage Monitor is checked using the machine-language FREQ program.[10]

Here are some things to keep in mind about the way temperatures are calculated in this program:

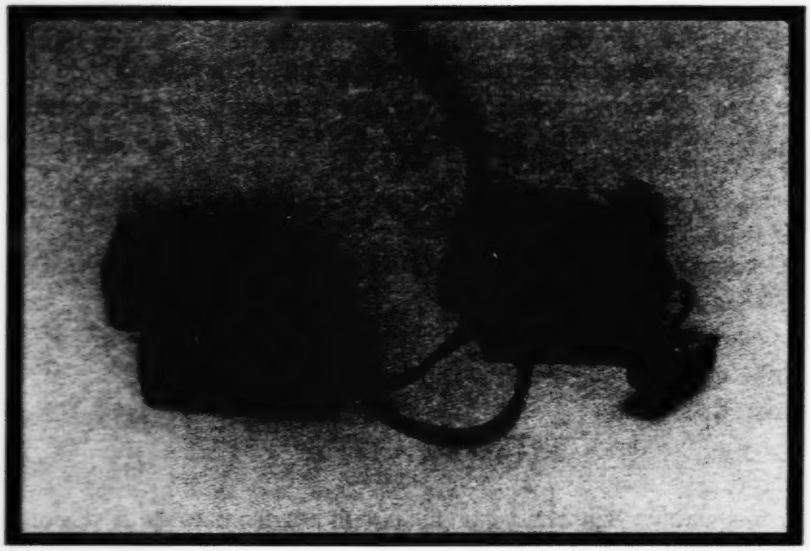
1. The calculation assumes that type K thermocouple wire is used. If any other type of thermocouple wire is used, the proportionality constant (24,400°C/volt) would be different.
2. The calculation also depends on the resistors used in the op amp circuit.
3. Before this method of calculating temperature can be used, the offset of the op amp (the 20-kΩ potentiometer) must be adjusted to 0.05 volts when the two ends of the thermocouple are at the same temperature.
4. The resistors in the op amp circuit were selected to give the proper gain so that the output of the system covers the voltage range of the Voltage Monitor. The maximum thermocouple voltage for type K thermocouple wire is about 0.060 volts. This yields a maximum voltage out of the amplifier of 46.5 x 0.60 + 0.05 = 2.84 volts.

---

[9] You did this in Step #6 of the "Step-by-Step Instructions."

[10] For details on how the Voltage Monitor reads a voltage, refer to the "Background Information" and "Program Design Notes" sections of the *Voltage Monitor* project.

PROJECT 9:
pH METER



The effective acidity or alkalinity of a solution is expressed as its pH. Chemists often measure pH using *pH electrodes*, which produce a small voltage that is proportional to pH. This project consists of building an amplifier circuit that will allow the signal from a pH electrode to be measured by the Voltage Monitor that you built in Project 7. The PH program on the *Project Programs* disk allows you to calibrate the system and displays the pH.

PROJECT REQUIREMENTS

Project Prerequisite

The Voltage Monitor from Project 7 is needed to build the pH Meter.

Background Knowledge

- General: If you have never worked with pH electrodes, you should seek some information on the proper method of using and storing the electrode. Refer to electrode manuals or consult a chemistry teacher. pH electrodes are fragile and expensive; use them carefully.

- Electronics: Experience with soldering is recommended. Specific component placement on the circuit board is not given for this project, so you have to be able to read a simple schematic diagram and decide how to position the components.

- Programming: None required for the core project; some of the project extensions require experience with BASIC programming.

Parts

☐ pH electrode (Almost any "combination" electrode can be used. An electrode with a BNC connector is the easiest to use, because the connector for this type of electrode is available at Radio Shack stores. pH electrodes are sold by scientific-apparatus supply houses. We have had good luck using a pHoenix 5733526-03B Combination pH Electrode sold by

Numberg Scientific, 6310 S.W. Virginia, Portland, OR 97201 (503) 246-8297. It sells for approximately $45.00.)

☐ connector for connecting pH electrode to the amplifier circuit (If you are using an electrode with a BNC connector, you can use a Radio Shack Type 1094 BNC [RS #278-105].)

☐ BIFET input op amp: TL041 (a Texas Instruments part available at Jameco) or LM356 (a National Semiconductor part available at Digi-Key) (It is essential that an op amp with extremely high input impedance be used, such as one of these two.)

☐ cable with 5-pin DIN connector (This is used for connecting the pH amplifier to the Voltage Monitor. Use one-half of a DIN patch cord [RS #42-2151] or a 5-pin DIN plug [RS #274-003] with 2 wires soldered to it.)

☐ circuit board (Any small circuit board designed for use with DIP ICs can be used. One possibility is one-half of a RS #276-159.)

☐ 1-MΩ, PC-board potentiometer (RS #271-229)

☐ 8-pin IC socket (RS #276-1995)

☐ 0.1-μF capacitor, 10 volt (RS #272-1053 or RS #272-1432)

☐ 100-kΩ resistor, 1/4-watt

☐ 33-kΩ resistor, 1/4-watt

☐(2) 9-volt battery connectors (RS #270-325)

☐(2) 9-volt batteries

☐ miscellaneous wire for connections on the circuit board and to the BNC connector

Tools and Supplies

Soldering iron, solder, wire cutters, wire strippers, digital DC voltmeter, distilled water, pH 7 buffer solution and two other buffer solutions.

BACKGROUND INFORMATION

One of the most important measurements made in chemistry is of pH. The pH scale is a convenient way of expressing the relative acidity or alkalinity of a solution. It is a measure of the hydrogen-ion concentration of the solution.[1] The pH scale extends from 1 to 14. The pH of pure water is 7.00. A solution with a pH lower than 7 is an acid; the lower its pH, the stronger the acid. If the pH is higher than 7, the substance is a base.

There are two common methods of determining pH. When great accuracy is not required, chemicals that change color with pH are used. Paper strips that contain pH sensitive chemicals are often used in chemistry classes. Some of these pH test strips take on a color that gradually changes with pH and a color chart is used to estimate the value. Other test strips simply change from one color to another at a specific pH.

When a more accurate measurement of pH is needed, an electrochemical method of pH measurement is used. In these measurements pH electrodes are used. Most modern pH electrodes used in chemistry classes are *combination* electrodes. They are actually two electrodes combined in one small tube. The combination electrode can be compared to a battery. It produces a small voltage between its two electrodes when it is placed in a solution. The voltage produced depends on the pH of the solution. The *Nernst equation* gives the relationship between the voltage produced and the pH:[2]

---

[1] The pH of a solution is the negative of the logarithm of the H^+ ion concentration. Refer to a chemistry text for information about acids, bases, and the pH scale.

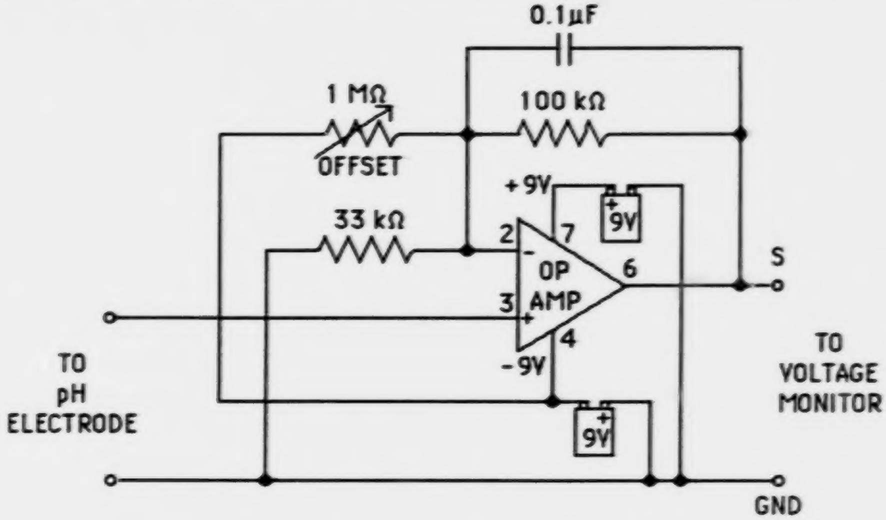
[2] For more information on the Nernst equation, consult a college-level chemistry text or *Van Nostrand's Scientific Encyclopedia* under "Titration, Potentiometric."

$$V = V_0 - 59\ \text{mV} \times \text{pH}$$

The voltage produced changes by 59 millivolts for each pH number.[3] As the pH increases, the voltage decreases. V_0 is a constant voltage that depends on the reference electrode used. The nice thing about this relationship is that the voltage varies in a simple, linear way with pH. As a result, the voltage can easily be used as a way to measure pH indirectly.

The traditional way to use a pH electrode is with a *pH meter*. A pH meter consists of a high-quality amplifier and a method of displaying the reading. The amplifier in the meter must have *high impedance*. This means that it must draw very little current from the pH electrode. The amplifier increases the small voltage from the electrode to a range at which it can be used to operate the display. The display is usually a meter with a moving needle indicating pH or a digital display.

In this project we will build our own amplifier to produce a voltage signal to feed into the Voltage Monitor and then use the computer as the display device. The high-impedance amplifier needed can be made from just a few inexpensive components. A schematic of the amplifier is shown below:



PH AMPLIFIER SCHEMATIC

This circuit uses two 9-volt batteries for its power supply.[4] The amplifier has a *gain* of four, which means that the output voltage is four times the input voltage. The voltage out will then vary by 4 x 59 mV or 0.24 volts for each pH number change. The circuit also includes an *offset potentiometer* that allows the output voltage to be adjusted up or down. We will use this potentiometer to set the output voltage at 1.75 volts when the pH electrode is in a solution with a pH of 7.00. This setting and the gain of four were chosen to yield a good range of output voltages to be fed into the Voltage Monitor. If the pH is changed by 7 pH numbers, the voltage out of the amplifier will change by 7 x 0.24 V = 1.7 V. Therefore, the maximum voltage possible from the amplifier is 1.75 V + 1.7 V = 3.45 V. The minimum voltage possible is 1.75 V - 1.7 V = 0.05 V. This range of voltages is within the 0 to 3.5 volt range of the Voltage Monitor.

---

[3] This is the Nernst equation for solutions at 25°C. The constant, which is shown here as 59 mV, changes with the temperature of the solution. The constant is proportional to the temperature of the solution in kelvins.

[4] The other amplifier circuits in this book (Projects 8 and 10) are powered using the +5V line from the Apple game port. This pH circuit must have both negative and positive power supply leads, because the voltage signal from the electrode can be either negative or positive.

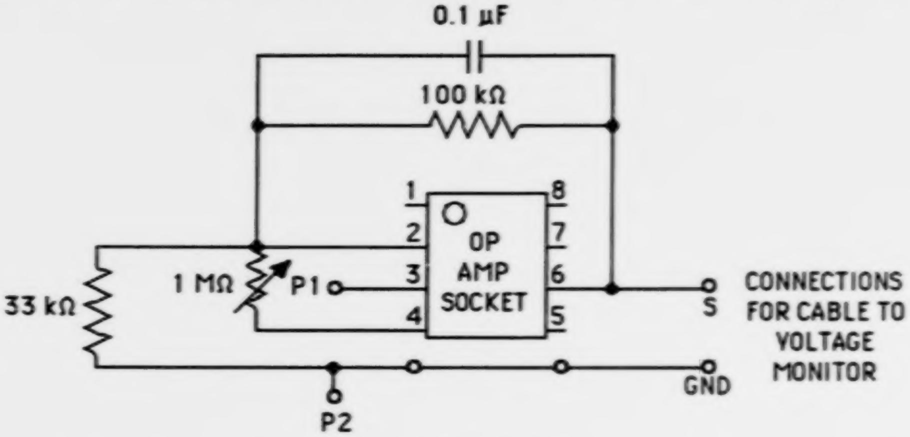
A program is provided on the *Project Programs* disk that will convert voltage readings from the Voltage Monitor to pH values. This program also includes a CALIBRATION mode. This mode allows you to set the exact values of the constants used in the calculation of pH from voltage so that your system reads accurately. You do this by using two *buffer* solutions. Buffers are solutions made especially to remain at a particular pH.[5] To use the CALIBRATION mode, you place the electrode in the first buffer and allow the voltage reading to stabilize. You then enter the pH of the buffer solution. Rinse the electrode in distilled water and repeat this procedure using the second buffer solution. A file named PH.CAL stores the proper constants to make the correct conversion from voltage to pH.

A good, free source of information on pH measurement is the *Complete pH and Conductivity Measurement Handbook and Encyclopedia* from Omega Engineering, One Omega Drive, P.O. Box 4047, Stamford, CT 06907, (203) 359-1660. This catalog of pH measurement equipment also includes several pages of useful information on using and maintaining pH electrodes.

STEP-BY-STEP INSTRUCTIONS

☐ 1. **Solder the 8-pin socket onto the circuit board.** Place the socket near the center of the nonplated side of the circuit board. One of the corners of the socket will be marked with a notch or other mark to indicate that this is where the #1 pin of the IC should be placed. The other holes of the socket are referred to by number. They are numbered in a counterclockwise pattern as you view the socket from the top (with the pins of the socket pointing away from you). Solder all 8 of the pins to the plated side of the circuit board.

☐ 2. **Assemble the amplifier circuit.** Solder the two resistors, one capacitor and one potentiometer to the leads of the socket. The working diagram below shows the proper positioning of the components. Note that only two of the three leads on the potentiometer will be used—the slider (middle) and either one of the outside leads. You may have to ream out the holes on the circuit board for the potentiometer leads. Select an area of the circuit board away from the IC socket with five electrically-connected holes to use as a GND terminal. Use short jumper wires to make connections as you need them. Leave holes at the two locations labeled P1 and P2 for the pH electrode connections. Leave holes for the two leads of the cable to the Voltage Monitor (S and GND).



☐ 3. **Add the battery connectors to the circuit.** Solder the two battery connectors as shown in the diagram below. Carefully note the orientation of the "+" and "-" leads. On battery connectors

---

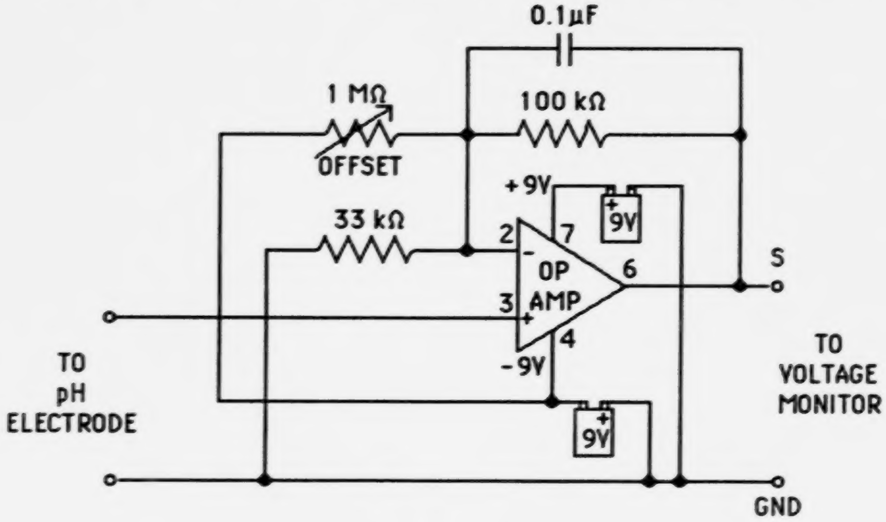
[5] Liquid buffers and capsules that can be used to make up buffer solutions are available from scientific-equipment companies.

$$V = V_0 - 59\ \text{mV} \times \text{pH}$$

The voltage produced changes by 59 millivolts for each pH number.[3] As the pH increases, the voltage decreases. V_0 is a constant voltage that depends on the reference electrode used. The nice thing about this relationship is that the voltage varies in a simple, linear way with pH. As a result, the voltage can easily be used as a way to measure pH indirectly.

The traditional way to use a pH electrode is with a *pH meter*. A pH meter consists of a high-quality amplifier and a method of displaying the reading. The amplifier in the meter must have *high impedance*. This means that it must draw very little current from the pH electrode. The amplifier increases the small voltage from the electrode to a range at which it can be used to operate the display. The display is usually a meter with a moving needle indicating pH or a digital display.

In this project we will build our own amplifier to produce a voltage signal to feed into the Voltage Monitor and then use the computer as the display device. The high-impedance amplifier needed can be made from just a few inexpensive components. A schematic of the amplifier is shown below:



PH AMPLIFIER SCHEMATIC

This circuit uses two 9-volt batteries for its power supply.[4] The amplifier has a *gain* of four, which means that the output voltage is four times the input voltage. The voltage out will then vary by 4 x 59 mV or 0.24 volts for each pH number change. The circuit also includes an *offset potentiometer* that allows the output voltage to be adjusted up or down. We will use this potentiometer to set the output voltage at 1.75 volts when the pH electrode is in a solution with a pH of 7.00. This setting and the gain of four were chosen to yield a good range of output voltages to be fed into the Voltage Monitor. If the pH is changed by 7 pH numbers, the voltage out of the amplifier will change by 7 x 0.24 V = 1.7 V. Therefore, the maximum voltage possible from the amplifier is 1.75 V + 1.7 V = 3.45 V. The minimum voltage possible is 1.75 V - 1.7 V = 0.05 V. This range of voltages is within the 0 to 3.5 volt range of the Voltage Monitor.

---

[3] This is the Nernst equation for solutions at 25°C. The constant, which is shown here as 59 mV, changes with the temperature of the solution. The constant is proportional to the temperature of the solution in kelvins.

[4] The other amplifier circuits in this book (Projects 8 and 10) are powered using the +5V line from the Apple game port. This pH circuit must have both negative and positive power supply leads, because the voltage signal from the electrode can be either negative or positive.

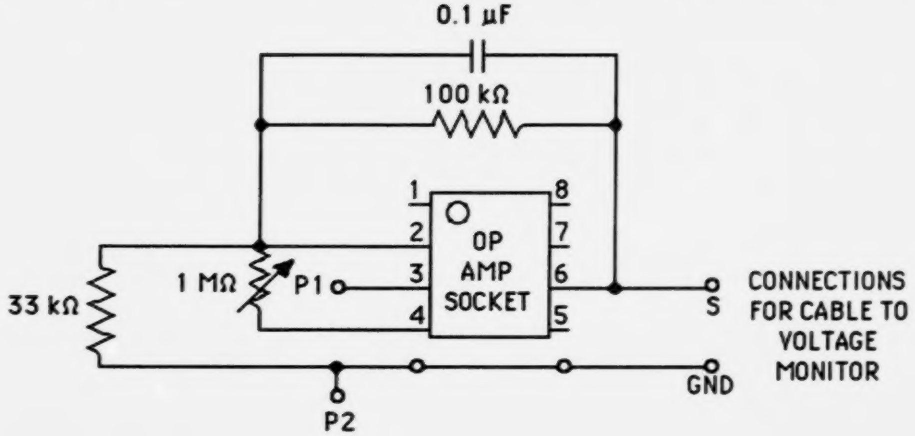
A program is provided on the *Project Programs* disk that will convert voltage readings from the Voltage Monitor to pH values. This program also includes a CALIBRATION mode. This mode allows you to set the exact values of the constants used in the calculation of pH from voltage so that your system reads accurately. You do this by using two *buffer* solutions. Buffers are solutions made especially to remain at a particular pH.[5] To use the CALIBRATION mode, you place the electrode in the first buffer and allow the voltage reading to stabilize. You then enter the pH of the buffer solution. Rinse the electrode in distilled water and repeat this procedure using the second buffer solution. A file named PH.CAL stores the proper constants to make the correct conversion from voltage to pH.

A good, free source of information on pH measurement is the *Complete pH and Conductivity Measurement Handbook and Encyclopedia* from Omega Engineering, One Omega Drive, P.O. Box 4047, Stamford, CT 06907, (203) 359-1660. This catalog of pH measurement equipment also includes several pages of useful information on using and maintaining pH electrodes.

STEP-BY-STEP INSTRUCTIONS

☐ 1. **Solder the 8-pin socket onto the circuit board.** Place the socket near the center of the nonplated side of the circuit board. One of the corners of the socket will be marked with a notch or other mark to indicate that this is where the #1 pin of the IC should be placed. The other holes of the socket are referred to by number. They are numbered in a counterclockwise pattern as you view the socket from the top (with the pins of the socket pointing away from you). Solder all 8 of the pins to the plated side of the circuit board.

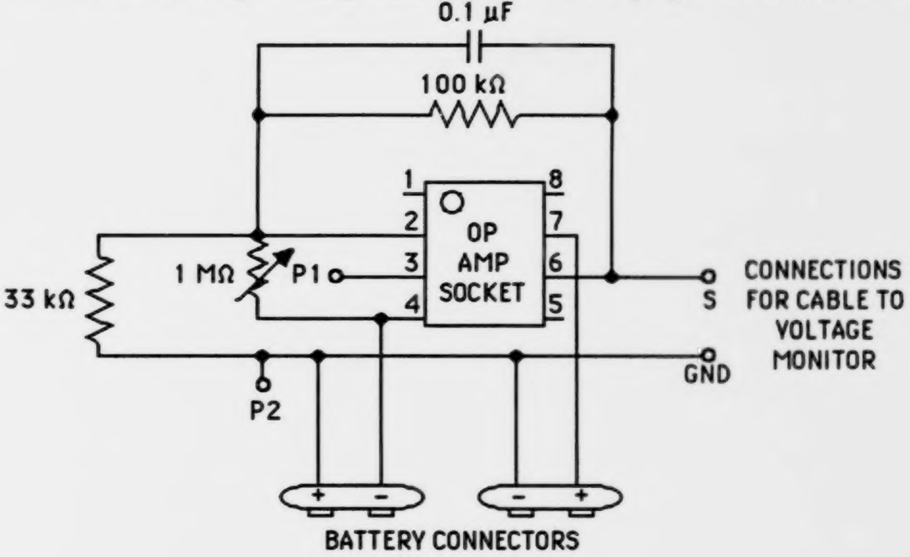
☐ 2. **Assemble the amplifier circuit.** Solder the two resistors, one capacitor and one potentiometer to the leads of the socket. The working diagram below shows the proper positioning of the components. Note that only two of the three leads on the potentiometer will be used—the slider (middle) and either one of the outside leads. You may have to ream out the holes on the circuit board for the potentiometer leads. Select an area of the circuit board away from the IC socket with five electrically-connected holes to use as a GND terminal. Use short jumper wires to make connections as you need them. Leave holes at the two locations labeled P1 and P2 for the pH electrode connections. Leave holes for the two leads of the cable to the Voltage Monitor (S and GND).



☐ 3. **Add the battery connectors to the circuit.** Solder the two battery connectors as shown in the diagram below. Carefully note the orientation of the "+" and "-" leads. On battery connectors the "+" lead is usually red; the "-" lead is usually black. The "+" lead from one connector and the "-" lead from the other connector go to the ground terminal. Do not plug in the batteries at this time.

---

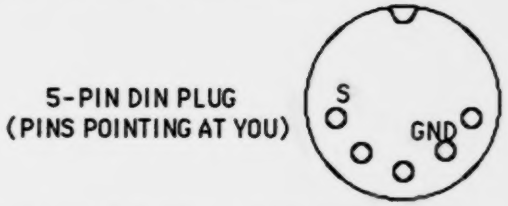
[5] Liquid buffers and capsules that can be used to make up buffer solutions are available from scientific-equipment companies.



WORKING DIAGRAM OF THE PH AMPLIFIER CIRCUIT (TOP VIEW)

☐ 4. **Attach the connector for the pH electrode.** Prepare two short (about 10 cm) pieces of wire. Strip both ends of both wires. Connect one of the wires from the point labeled P1 on the diagram above to the "center" terminal on the BNC connector. Connect the other wire from P2 to the "outside" terminal of the BNC connector.

☐ 5. **Connect the cable to the Voltage Monitor.** Make up a cable to extend from the pH amplifier circuit to the Voltage Monitor (Project 7). If you followed our suggestion in constructing the Voltage Monitor, you will have a 5-pin DIN socket on the box for this connection. For the cable, you need 2 conductors ending in a 5-pin DIN plug. Two suggestions for this wire are: (1) Use a 5–pin DIN plug and solder on two wires as shown below:[6]



or (2) Buy a 6-foot DIN patch cord that has a 5-pin DIN plug on each end and cut it in half. This gives you two ready-to-use cords. Use an ohmmeter to identify the correct leads on the plug. The extra cord can be used for another project.[7]

Solder the two leads to the points labeled "S" and "GND" on the working diagram of the circuit.

---

[6] The other projects that use DIN plugs have 3 pins connected. This project does not use the +5V power supply lead from the computer as the others do.

[7] Two other projects in this book that require a similar cord are *Thermocouple* (Project 8) and *Strain Gage* (Project 10).

☐ 6. **Plug in the op amp.** Plug the op amp IC into the 8-pin socket. Be sure pin #1 is oriented at the proper corner of the socket. Pin #1 is marked with a circular hole in the top of the IC.

☐ 7. **Connect the batteries to the circuit.** Plug both of the 9-volt batteries into their connectors. At this point, look, feel and smell for any overheating of the op amp. If the op amp gets warm, you have a wiring error. Disconnect the batteries and double-check your circuit.

☐ 8. **Test the amplifier without the computer and adjust the offset.** Connect a digital voltmeter with a 0 - 9 VDC range to the output lines of the amplifier circuit. The positive voltmeter lead should be attached to the "S" line. The negative lead should be attached to the GND lead. Connect the pH electrode to the circuit. Place the pH electrode in a pH 7 buffer solution. Monitor the voltage with the meter. Try adjusting the 1-MΩ potentiometer. The voltage read by the meter should change as you adjust the potentiometer. If it does not, refer to the "Troubleshooting" section below. If the voltage can be adjusted with the potentiometer, set the potentiometer so that the voltage is about 1.75 volts when the pH electrode is in the pH 7 buffer.

☐ 9. **Try out the circuit with the computer.** With the computer off, connect the Voltage Monitor to the game port. Plug the pH amplifier circuit into the Voltage Monitor. Turn on the computer and run the PH program on the *Project Programs* disk. Select mode P ("READ PH"). The program should display the approximate pH of the solution in which the electrode is placed.[8] Try several different solutions. If the reading does not change, or if the pH value gets lower when it should get higher, refer to the "Troubleshooting" section below. If the reading changes but is inaccurate, you simply need to calibrate the system.

☐ 10. **Calibrate the system.** For accurate calibration, you will need two pH buffer solutions. Run the PH program and select "CALIBRATION" from the main menu. Place the pH electrode in one buffer and allow the reading to stabilize. Press <RETURN> and then enter the pH of the buffer being used. Now rinse off the pH electrode and move it to a second buffer. After the reading stabilizes, press <RETURN> and then enter the pH of the buffer. This creates a new file on the disk (PH.CAL) that will save the values needed to allow the PH program to calculate the pH from the voltage reading. This file will be used any time you run the PH program. It can be changed later by repeating the calibration procedure.

☐ 11. **Check the accuracy of the system.** Rinse off the electrode in distilled water and then place it in a third buffer solution. If the system is working well and calibrated correctly, the pH reading should closely match the labeled value of the buffer.

Disconnect the two batteries when you are not going to be using the pH amplifier for an extended period.[9]

TROUBLESHOOTING

• Test the pH electrode

pH electrodes can be cantankerous. If you are having problems with your pH measurement system, try to determine if the problem is with your electrode. If possible, test the electrode by using it with a commercial pH meter. Another way to test the electrode is to connect it to a good quality (high-impedance) digital voltmeter that can read millivolts.[10] The voltage across the two terminals of the pH electrode should vary as you change the pH of the solution. In fact, the voltage from the electrode should change by about 59 mV for each pH number.

---

[8] Notice that the program does not round off the pH at all. You are encouraged to add rounding to the program as a project extension.

[9] A possible project extension is to add a switch to allow you to disconnect the batteries from the circuit.

[10] This test will not work with inexpensive analog voltmeters, because the pH electrode does not produce enough current to operate them.

- **Visual circuit checks**

If the pH amplifier circuit is not working properly, check for these most common errors:

◊ Is your op amp chip inserted properly? Pin #1 is marked with a dot. Is it where it should be?
◊ Have you made the connections from the Voltage Monitor correctly?
◊ Does your op amp get hot when you turn things on? Hook things up and turn off the computer. Touch the top of the op amp. If it is hot, you must have a serious wiring problem and you may even need to replace the IC after you fix the wiring error.
◊ Visually double-check all of the connections on the op amp. In particular, make sure you have not mixed up the two resistors. Make certain the power supply leads from the battery holders are wired correctly.

- **Make sure your buffer solutions have not deteriorated**

Buffer solutions have a limited shelf life. It is possible that they are not the pH that they are labeled.

- **Check the Voltage Monitor**

Make sure the problem is not in your Voltage Monitor. Connect a known voltage source, such as a 1.5-volt battery, to the Voltage Monitor and test it using the VOLTAGE program. Remember that the maximum voltage that can be used with the Voltage Monitor is 3.5 volts.

- **Check the circuit with a voltmeter**

Connect the amplifier circuit to the Voltage Monitor. Plug the Voltage Monitor into the game port and turn on the computer. Set the voltmeter on a range to read DC voltages between 0 and 5 volts and connect the negative lead of the voltmeter to the GND terminal on the circuit board. Check the voltage at the following pins of the op amp:

PIN #	EXPECTED VOLTAGE	COMMENTS
7	+7 V to +9 V	Positive power-supply lead from one battery. If you do not have a large positive voltage here, check your connections to that battery, or check the battery itself.
4	-9 V to -7 V	Negative power-supply lead from one battery. If you do not have a large negative voltage here, check your connections to that battery, or check the battery itself.
3	-0.4 V to +0.4 V	This is the input voltage from the pH electrode. If you are using a good quality digital voltmeter, you should be able to read the voltage from the pH electrode here. If the voltage at pin #3 does not vary as you change the pH of the solution, the electrode may not be working, or is not connected properly.
6	+0.05 V to +3 V	This is the output voltage of the amplifier. It should change by 0.24 volts for each pH unit change. This voltage should also be affected by adjustment of the 1-MΩ potentiometer. If the potentiometer has been set properly, this voltage should be about 1.75 volts when the electrode is in pH 7 buffer. If the voltage at this point does not vary as the potentiometer is adjusted, there may be an error in the wiring of the offset-circuit potentiometer. Make sure the 1-MΩ resistor is connected to the negative power supply (pin #4 of the op amp). If you do get the expected response at this pin, then your pH amplifier is working properly and there must be a problem with the Voltage Monitor or its connections.

- **Special problems**

◊ pH values go the "wrong way"

This is not really a problem. It is caused by reversing the leads from the circuit board to the BNC connector, and can be corrected by recalibrating the probe. Follow the calibration procedure of the PH program and this problem will be solved. The constants used to calculate pH will be changed to correct for the way you have the pH electrode wired.

◊ pH readings are slow to change when you move the electrode to a different solution

This is one of the signs of a pH electrode that is not in good condition. Refer to the manufacturer's instructions on how to "rejuvenate" the electrode.

PROJECT EXTENSIONS

Circuit Modifications

◊ Place the pH amplifier circuit in a box

Select an experimenter box large enough to contain the circuit board and the two batteries. You may want to mount battery holders (RS #270-326) inside the box. Another feature you can add is a switch on the power-supply leads from the batteries. Use a double-pole switch to disconnect both batteries from the op amp when the amplifier is not being used.

Drill holes in the box for mounting the connector to the pH electrode and for the cable to the Voltage Monitor.

◊ Change to metal-film resistors on the circuit board

Standard carbon-composition resistors change resistance slightly as their temperature changes. Metal-film resistors can be substituted for the two resistors on the circuit board to reduce this effect.

◊ Connect the pH-amplifier circuit to a DC power supply

Use a power supply that can provide both positive and negative voltages. Use the power supply to take the place of the batteries.

◊ Add a voltage regulator to the circuit

The calibration of this measurement system may change slightly as the power-supply voltages change. This is bound to happen if you use batteries for the power supply, as we recommend in this project. The consistency of the pH system can be improved if precision voltage regulators are added to both the negative and positive power-supply lines. Several adjustable voltage-regulator ICs are sold by Radio Shack stores (RS #276-1778, RS #276-1740).

Program Improvements

The PH program on the *Project Programs* disk is a fairly simple program that can be improved by adding new features. The "Program Design Notes" section below explains how the PH program operates. It should help you in making modifications. Some suggested changes are given below:

◊ Change the program so that the pH displayed is rounded off to an appropriate number of significant digits.
◊ Add an option to display or print the table of pH and time.
◊ Add a mode to the program that displays the pH in large, low-resolution graphics block digits so they can be read at a distance.
◊ Add a mode to the program that graphs the pH readings versus time.

◊ Add a mode that allows you to collect and graph data for doing titrations. The data collected should be pH versus the amount of titrate added during the titration. Graphs of this type are frequently used in chemistry.

Additional Experiments

◊ Test the pH of common substances

Experimentally determine the pH of the following materials. Compare your results with values published in reference books.[11]

Fresh water
Seawater
Soil
Plant tissues and fluids
Blood
Urine

Soft Drinks
Baking Soda
Vinegar
Milk
Lemon Juice
Fruit Jellies

◊ Perform a titration

A common procedure in chemistry is a *titration*. In a titration, an acid is carefully added to a base (or vice versa). The concentration of the solution added is known and it is referred to as the *standard solution*. Standard solution is added very gradually, and the volume added is carefully noted. At the same time, the pH of the combined liquids is monitored. The *endpoint* of the titration is determined when the pH readings go through a rapid change. Your pH measurement system is an ideal way of identifying the endpoint. The addition of a pH graphing mode to the PH program can allow titration curves to be generated by the computer.

◊ Study the change in pH of distilled water exposed to air

Carefully calibrate the pH measurement system. Place the electrode in distilled water. The pH should be 7.00. Let the distilled water stand in an open container for several hours. Check the pH periodically (or modify the PH program so that it does this automatically). You will probably notice a gradual change in the pH of the water. What causes it?

◊ Use the pH amplifier system with specific-ion electrodes

Specific-ion electrodes may be used in the same way as pH electrodes. These electrodes produce a voltage that varies with the concentration of a particular ion (for example, chlorine). These electrodes produce a voltage proportional to the concentration of a particular ion.

SUPPLEMENTAL INFORMATION

Program Design Notes

The PH program on the disk will do the calculations necessary to convert the voltage readings from the amplifier to pH readings. Here is a section-by-section explanation of the program.

---

[11] A list of pH values for various substances appears in *Van Nostrand's Scientific Encyclopedia* and the *Handbook of Chemistry and Physics.*

Lines 100 to 990: Initialization

This section loads the machine-language FREQ program.[12] It then reads the PH.CAL file, which gives the program the conversion constants needed to calculate pH from voltage. It also displays the main menu.

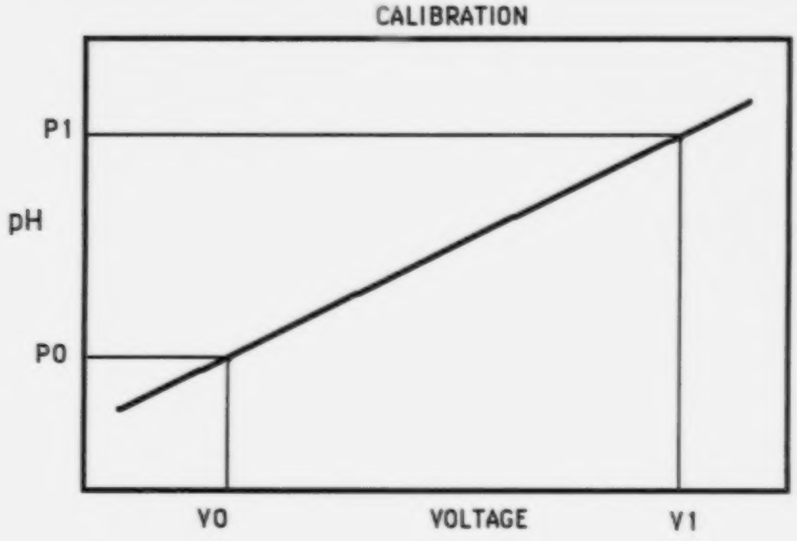
Lines 1000 to 2990: Display pH
This section simply reads the pH and displays it on the video monitor.

Lines 3000 to 3370: Calibration
This section allows you to calibrate the pH measurement system. First, place the pH electrode in a buffer solution. Allow the voltage to stabilize and then press the <RETURN> key. Enter the pH of the buffer. Next, rinse the electrode and place it in a second buffer. Let the reading stabilize, press <RETURN>, and enter the pH of the second buffer. When you enter these quantities, the program will save the calibration data on the disk in the file PH.CAL. This data will then be used each time the program starts.

Lines 10000 to 10990: Voltage check subroutine
These lines are used to check the voltage from the Voltage Monitor. Refer to the "Program Design Notes" in Project 7 for details.

How the Voltage is Converted to a pH

The graph below shows how the pH is calculated. The output voltage of the pH amplifier is assumed to be related to the pH in a linear way. The two points on the graph are established using the CALIBRATION mode. P0 and P1 are the pH of the two buffers used. V0 and V1 are the corresponding voltages.



CALIBRATION GRAPH OF THE PH PROGRAM

The slope of the line between these two points is calculated from:

slope = (P1-P0) / (V1-V0)

The slope and the values of P0 and V0 are saved in the PH.CAL file.

---

[12] FREQ is also used with the Voltage Monitor. It is discussed in detail in the "Program Design Notes" for Project 7.

To convert a voltage reading (V) to a pH value, the program "interpolates" the value along the line on the graph by calculating:

$$pH = P0 + slope \times (V - V0)$$

The variables used in the program PH are as follows:

C	Counts from the Voltage Monitor
D$	Control D, used for disk access
PH	Current pH reading
P0	pH of first buffer used in calibration
P1	pH of second buffer used in calibration
M$	Mode of operation
SL	Slope
V	Present voltage reading
V0	Voltage reading with first buffer
V1	Voltage reading with second buffer

PROJECT 10: STRAIN GAGE

This project consists of building an apparatus that will allow you to measure force or weight. Electrical-resistance strain gages and an amplifier are used to produce a voltage that is proportional to the force to be measured. The voltage can be fed into the Apple using the Voltage Monitor from Project 7. Depending on how you set things up, the system can be used to measure very small forces (like a laboratory balance) or to weigh objects as heavy as people or cars. This system for measuring force is very similar to the method used in most electronic scales.

PROJECT REQUIREMENTS

Project Prerequisite

The Voltage Monitor (Project 7) is needed for this project.

Background Knowledge

- Electronics: This is one of the most challenging projects in the book. In general, the instructions are less specific than in other projects, and more of the details of how to do things are left to you. You should not attempt this project until you have had some experience assembling other electronic circuits (for example, other projects in this book). Knowledge of Ohm's law and some familiarity with electrical terminology are also recommended.
- Programming: None required for the core project. Some of the project extensions require knowledge of BASIC.

Parts

☐(2) 120-Ω strain gages (Two possibilities are the CEA-XX-240UZ-S120 from Measurements Group, Inc., P.O. Box 27777, Raleigh, NC 27611, (919) 365-3800, or the HBM 6/120LY11 from Omega Engineering, One Omega Drive, P.O. Box 4047, Stamford, CT 06907, (203) 359-1660.)

☐ metal bar onto which the strain gages will be mounted

- ☐ TLC274 quad op amp (This op amp is specifically designed for use with single 5-volt power supplies. It is manufactured by Texas Instruments and is sometimes available as RS #276-1750. If you cannot get the TLC274, you can do this project with an LM324 quad op amp [RS #276-1711].)
- ☐ 20-kΩ, multiturn potentiometer (RS #271-340)
- ☐ 5-kΩ potentiometer, single turn, PC mount (RS #271-217)
- ☐ shielded 3-conductor cable (3 conductors inside a metal shield)
- ☐ cable with 5-pin DIN plug (one-half of a DIN patch cord [RS #42-2151] or a 5-pin DIN plug [RS #274-003] with 3 wires soldered onto it)
- ☐ circuit board (two possibilities are RS #276-168 or RS#276-162)
- ☐ 14-pin IC socket (RS #276-1999)
- ☐(2) 100-Ω metal-film resistors, 1/4-watt, 1% (These resistors and the others needed for this project are included in the Radio Shack Metal-Film Resistor Assortment [RS #271-309])
- ☐(4) 100-kΩ metal-film resistors, 1/4 watt, 1%
- ☐(2) 4.7-kΩ metal-film resistors, 1/4 watt, 1%
- ☐ miscellaneous wire for connections on the circuit board

Note: A "Strain Gage Parts Kit" containing all of the components needed for this project, except the metal bar, is available from Vernier Software. This kit sells for $30.00. An order form is provided at the back of this book.

Tools and Supplies

Low-power soldering iron, solder, wire cutters, wire strippers, plastic electrical tape, glue, drill and drill bits.

BACKGROUND INFORMATION

Measurement of force with a computer is a little more difficult than most other measurements. Devices that produce an electrical signal in response to force are sold commercially as *force transducers*[1] and *load cells*. These devices are usually expensive. We will build an electronic circuit that will do the same job. The components are inexpensive, but the trade-off is that we will have to assemble a fairly complicated circuit.

We will use an indirect method to measure force. The force to be measured will be applied in such a way as to bend a metal bar. A basic principle of physics states that the amount a metal bar bends is proportional to the amount of force applied to the bar.[2] If we can measure the amount of bending of the bar, we have a way of determining the force applied.[3]

The device we will use to measure the bending of the bar is an *electrical-resistance strain gage*.[4] An electrical-resistance strain gage is a very fine foil grid that changes its resistance when its dimensions are changed. Many different types of electrical-resistance strain gages are manufactured. They are sold for varying prices, starting at a few dollars each. The figure below shows what a typical strain gage looks like at about ten times actual size. The grid is bonded to thin plastic film. Two leads are available to connect the gage to a circuit.

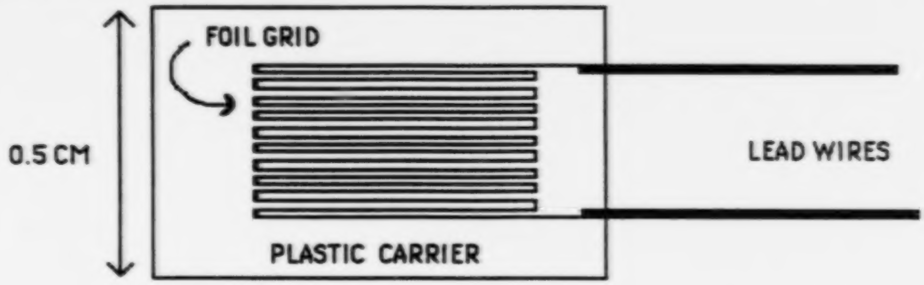
---

[1] A transducer is a device that produces an output signal when it responds to changes in some other signal.

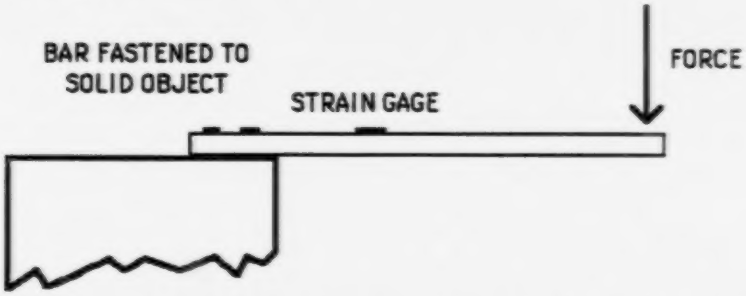
[2] This is known as Hooke's Law. It applies only to perfectly elastic materials and is not valid if the metal bar is bent beyond a limit where it is permanently distorted.

[3] In engineering terminology, *stress* is the force applied and *strain* is the bending. Specifically, strain is the deformation (bending) per unit length.

[4] You might expect this term to be spelled "gauge." In the electronics industry, the alternative spelling (without the "u") is used most often.



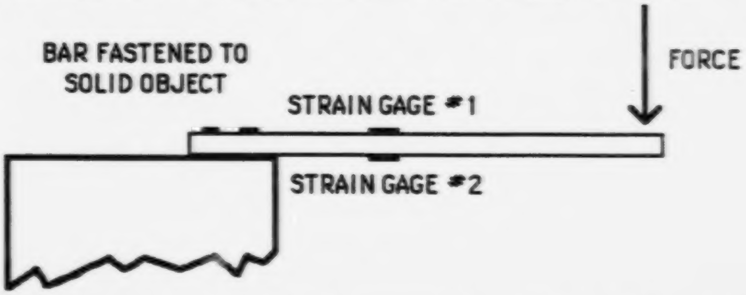
In the diagram below, a strain gage is glued to the top of a bar. The long sections of the wire are oriented parallel to the length of the bar.



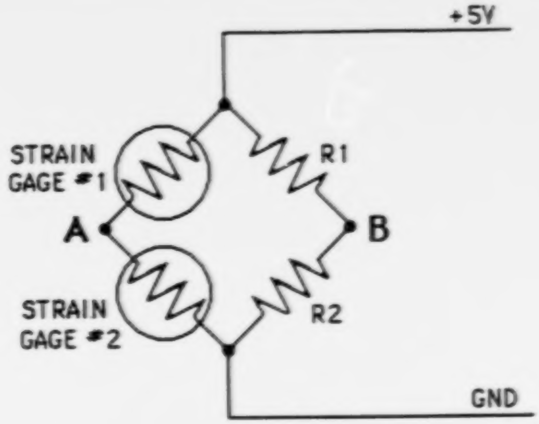
As the bar bends downward, the top of the bar stretches. The strain gage itself also stretches. The resistance of the strain gage increases, because the length of wire in the gage increases and the width of the wire decreases. This change in resistance is small, usually much less than 1%, but it can be measured accurately with the proper circuit. This is the heart of our system of measuring force.

The concept of measuring force by the bending of a metal bar can be used in a wide variety of situations. If you use a thin metal bar that bends very easily, the bar can measure very small forces. This could be the heart of an "electronic balance" that can weigh small objects. The other extreme would be to use a very large steel beam. With strain gages mounted on a large beam, objects as heavy as people or cars can be weighed.

Here is the specific arrangement we will use to measure the bending of the bar:



Two strain gages are used. As the bar bends downward, the resistance of the top gage will increase, while the resistance of the bottom one decreases. These two strain gages are connected electrically to make up one-half of what is called a *Wheatstone bridge circuit*. This is a well-known circuit designed to measure resistances with high accuracy.



The left side of the circuit is made up of two fixed resistors of approximately the same resistance as the strain gages. A 5-volt *excitation voltage* is applied across the bridge. Assume that all four resistances in the bridge are of exactly the same value when the bar is unstressed. The voltage at both points A and B in the diagram above will be one-half of the excitation voltage. When the bar is bent so that the resistance of strain gage #1 increases and strain gage #2 decreases in resistance, what happens? The voltage at point A will decrease slightly because the voltage drop in the top strain gage is now larger than the voltage drop in the bottom strain gage. The voltage at point B has not changed and is still equal to one-half the excitation voltage. There is now a small voltage difference between points A and B. The voltage between these two points can be monitored and used as a measure of the amount of bending of the bar. If the bar is not bent very far, the voltage produced is approximately proportional to the force applied.

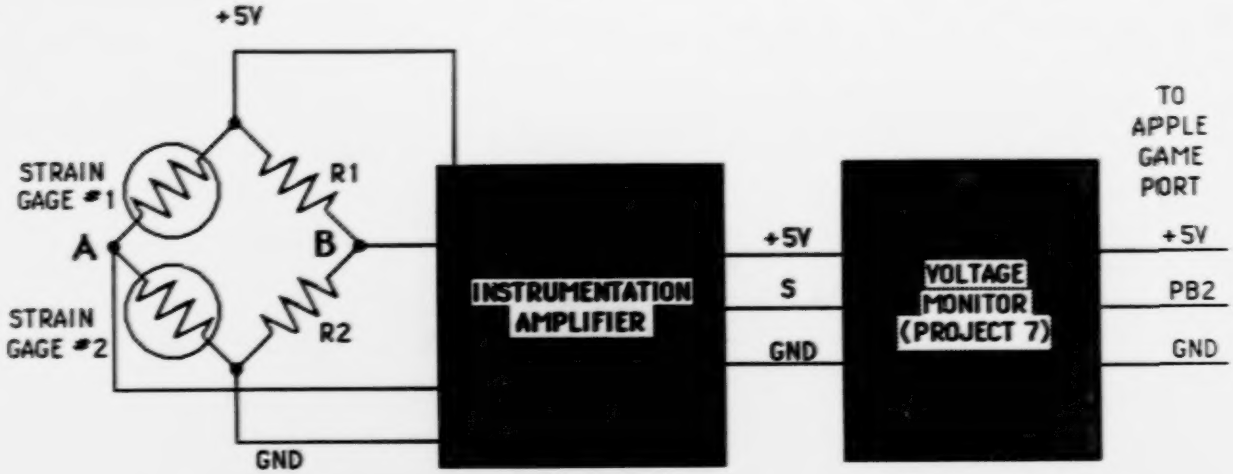
The voltage difference between points A and B is never very large—rarely as large as a millivolt. The next step is setting up a circuit that can amplify this very small voltage signal so that it can be read by the computer. The voltage needs to be amplified about 1000 times. This is referred to as a *gain* of 1000. We will build an *instrumentation amplifier* for this job. An instrumentation amplifier is designed to provide high-gain amplification of the difference in voltage of its two input signals. At the same time, it should be insensitive to other factors, produce very little "noise," and be very stable. Our instrumentation amplifier will be designed around a readily available and inexpensive integrated circuit—a quad op amp.[5]

Whenever an amplifier with a large gain is used, electrical noise can become a problem. This can happen with this project. If any unwanted voltage signal caused by stray magnetic or electric fields is present on the bridge circuit, it may be amplified along with the signal you want to study. For this reason, steps should be taken to minimize interference. These steps include keeping lead wires as short as possible, using shielded cables, and possibly housing the circuit in a metal box.

The last step in the force-measurement process is to measure the voltage out of the instrumentation amplifier. We will use the Voltage Monitor from Project 7 for this job. The programs provided on the *Project Programs* disk will allow you to calibrate the strain gage and display the force directly. The diagram below illustrates the force-measurement process:

---

[5] Another alternative, discussed in the "Project Extensions", is to use a commercial instrumentation amplifier.



This type of indirect force measurement is used frequently in engineering and in industrial testing and weighing systems. The system we will build will give you a reasonably accurate system for measuring forces. If you want to invest more time and money into measuring force, some suggestions for improving the circuit are given in the "Project Extensions."

There are some excellent free sources of information available on strain gages. Anyone working on this project should write to the two companies listed below for their technical information about strain gages. Both companies produce catalogs of their strain gages and accessories, but more importantly, the catalogs also include many pages of useful tips.

- Measurements Group, Inc., P.O. Box 27777, Raleigh, NC 27611, (919) 365-3800. Register for their "Technical Data Mailing Program," which includes product information, technical literature and a newsletter on using strain gages in schools. Also request copies of their Bulletins 307A and 309, which include information on their special line of student strain gages and their student manual for strain-gage technology.

- Omega Engineering, One Omega Drive, P.O. Box 4047, Stamford, CT 06907, (203) 359-1660. Request their *Pressure and Strain Measurement Handbook*. It includes a section that is virtually a textbook on how to use strain gages.

STEP-BY-STEP INSTRUCTIONS

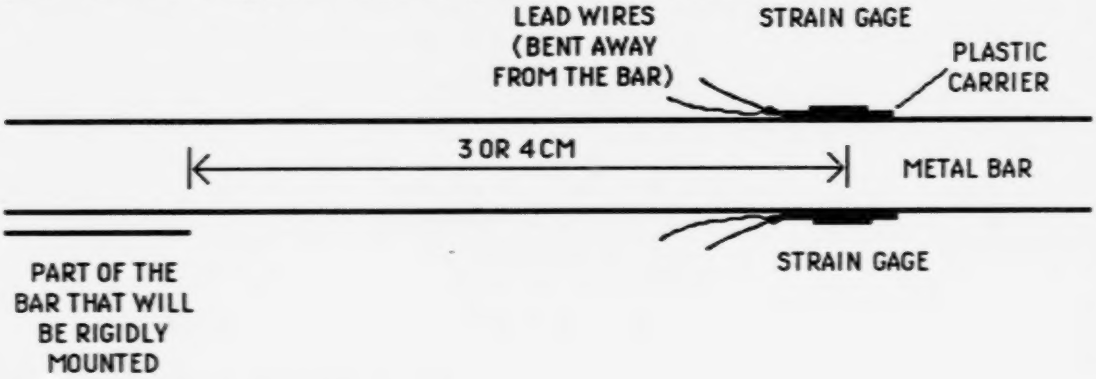
☐ 1. **Select the proper metal bar on which to mount the strain gages.** The bar you choose should flex visibly with the forces you plan to apply, but must not be permanently damaged under the maximum force you ever plan to apply. Use either steel, brass or aluminum. Remember that the amount of strain (bending) depends on where you apply the force. Take some care in deciding which bar to use, because once the strain gages are glued on, they cannot be removed.

☐ 2. **Mount the strain gages onto the metal bar.** The strain gages should be aligned with the longer, straight sections of the gage running along the length of the bar. The gages should be mounted about 3 or 4 cm from the point of support.[6] There are special adhesives sold for attaching strain gages, but household glues will work.[7] We recommend the cyanoacrylate (Superglue) type of adhesive.[8] The gage should be glued flat against the surface of the bar, with the plastic carrier side against the metal. Use a thin, even layer of glue between the gage and the bar.[9] The two metal leads of the strain gage must not touch the metal bar. Bend them away from the bar and put plastic electrical tape under them. The gage can be held in place with tape while the glue sets.

---

[6] The closer the gages are to the support, the more strain will be measured. If the gages are closer than about 3 cm, "edge effects" can complicate things.

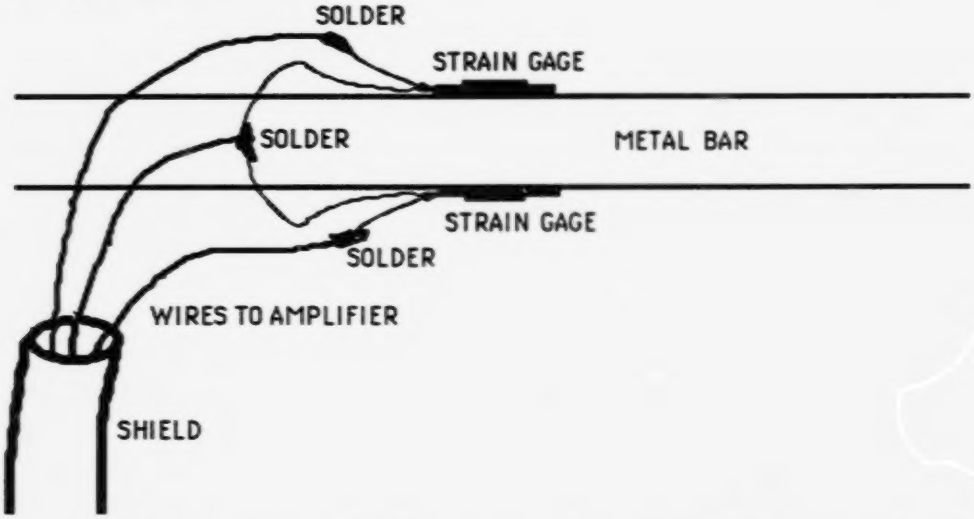
[7] Adhesives made especially for use with strain gages are sold by Omega Industries and Measurements Group.


LEAD WIRES
(BENT AWAY
FROM THE BAR)
STRAIN GAGE
PLASTIC
CARRIER
3 OR 4 CM
METAL BAR
STRAIN GAGE
PART OF THE
BAR THAT WILL
BE RIGIDLY
MOUNTED

☐ 3. **Assemble the bridge circuit.**

Note: Because the signal from the bridge is amplified over a thousand times, it is important to minimize electrical noise in the bridge circuit. Specific suggestions are given to help with this in the instructions below. In addition, keep all leads between the components of the bridge as short as possible. Any extra length of wire acts like an antenna, picking up unwanted electrical noise.

Carefully solder wires to the strain gages. Three wires will be needed. Use a shielded cable with three conductors inside. The advantage of the shielded cable is that it will help minimize the noise that can be picked up by the wires to the strain gage, which are acting as antennae. The wires from the strain gages will go to a circuit board that includes the instrumentation amplifier. Keep these wires fairly short (1 meter) by planning to place the amplifier near the strain gages. Solder the leads as shown in the diagram below:


SOLDER
STRAIN GAGE
SOLDER
METAL BAR
STRAIN GAGE
SOLDER
WIRES TO AMPLIFIER
SHIELD

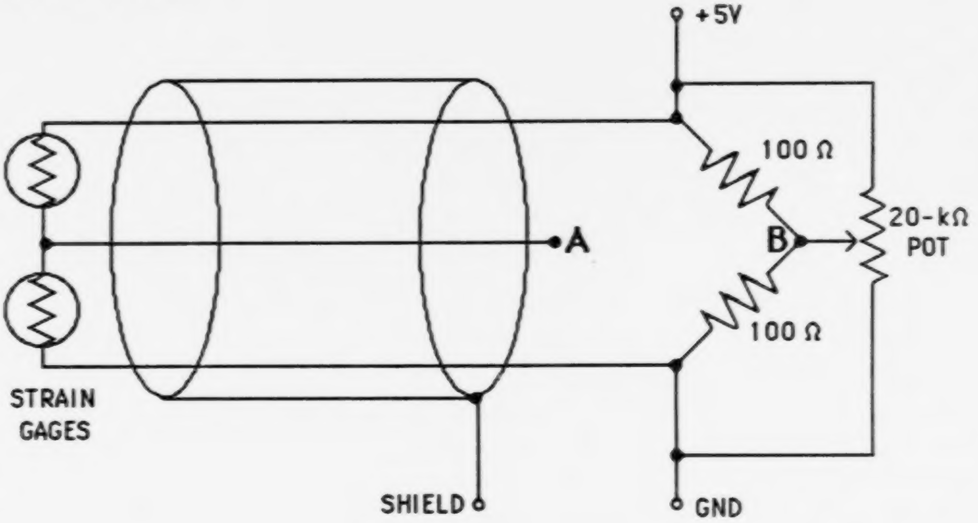
The leads on the strain gages are very fragile, so you need to make sure they are not ripped out by a pull on the wires. Tape the wire leads firmly to the metal bar so that if the wires are pulled, the pull is not transferred to the strain-gage leads. Make sure the metal of the strain gage leads does not touch the metal bar.

---

[8] Be careful about gluing your fingers with this adhesive.

[9] If the layer of adhesive between the metal and the gage is too thick, there will be a slow "gage creep" as the bar bends. The gage will change readings with time after a new force is applied.

Assemble the rest of the strain-gage bridge. The strain gages make up one side of the bridge. The other side is made up of two 100-Ω resistors. Metal-film resistors are recommended because they change resistance less than other resistors as the room temperature changes. The complete bridge circuit is shown below:



BRIDGE CIRCUIT

Mount the two 100-Ω resistors and the potentiometer on one side of a circuit board. Leave room on the other side of the board for the instrumentation amplifier (one 14-pin socket, 6 resistors and a potentiometer). The 20-kΩ potentiometer is used to "zero" the bridge. This allows you to adjust the "no stress" voltage at point B, so that it equals the voltage at point A.[10] The points labeled "A", "B", "+5V", "GND", and "SHIELD" will be connected later.

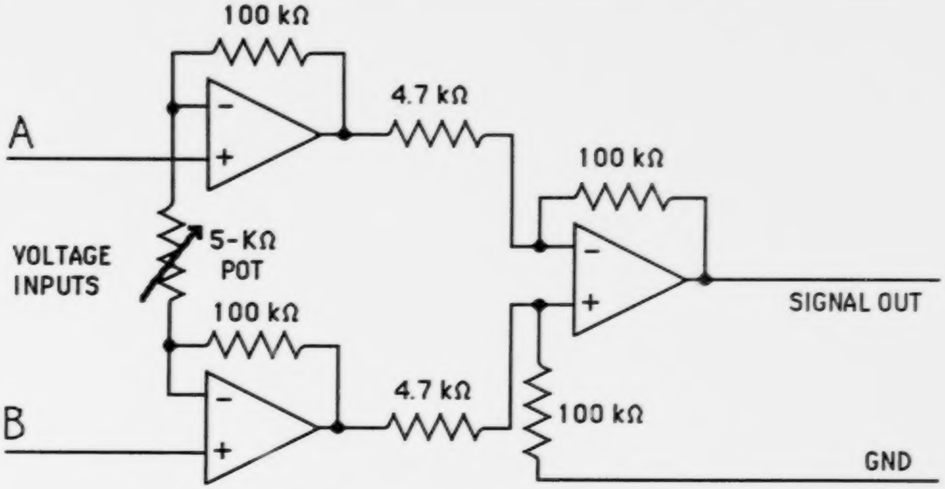
☐ 4. **Build the instrumentation amplifier.** The instrumentation amplifier shown below uses three op amps. Fortunately, integrated circuits are available that contain four op amps in one package. The recommended IC is the TLC274.[11] Other quad op amps can also be used.[12] The schematic diagram of the instrumentation amplifier is shown below:

---

[10] Actually, using this potentiometer the starting, "unstressed" output of the bridge can be set to any value you want. We will actually set this potentiometer so that the whole bridge/amplifier system gives us a slight positive voltage with no stress applied.

[11] The TLC274 is sensitive to static electrical discharges, so be careful when handling it.

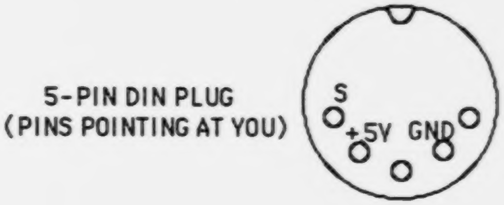
[12] This amplifier can also be built using the LM324 (RS #276-1711). The performance is slightly better using a TLC274.



INSTRUMENTATION AMPLIFIER SCHEMATIC

This circuit is a two-stage amplifier. The gain of the amplifier is set by adjusting the 5-kΩ potentiometer. The minimum gain is about 800 and the gain can be increased to several thousand. It is essential to use metal-film (1%) resistors for this amplifier.[13] A working diagram of the entire bridge/amplifier circuit is shown on the next page. Use this diagram as a guide to assemble your circuit.[14] Use a 14-pin socket for the integrated circuit and then insert the quad op amp after all soldering is completed. Be careful to insert the chip with pin #1 in the proper spot. Pin #1 is always marked with a dot or a notch.

☐ 5. **Prepare the cable to the Voltage Monitor.** Make up a 3-conductor cable to extend from the bridge/amplifier circuit to the Voltage Monitor (Project 7). If you followed our suggestion when constructing the Voltage Monitor, you will have a 5-pin DIN socket on the box for this connection. For the cable from the strain-gage amplifier, you need 3 conductors ending in a 5-pin DIN plug. Two suggestions are: (1) Use a 5-pin DIN plug and solder on three wires; or (2) Buy a 6-foot DIN patch cord that has a 5-pin DIN plug on each end and cut it in half.[15] This gives you two ready-to-use cords. The extra cord can be used for another project.[16] The three connections to use on the cord are shown below. Solder these leads to the points labeled "+5V", "S" and "GND" in the working diagram of the circuit.



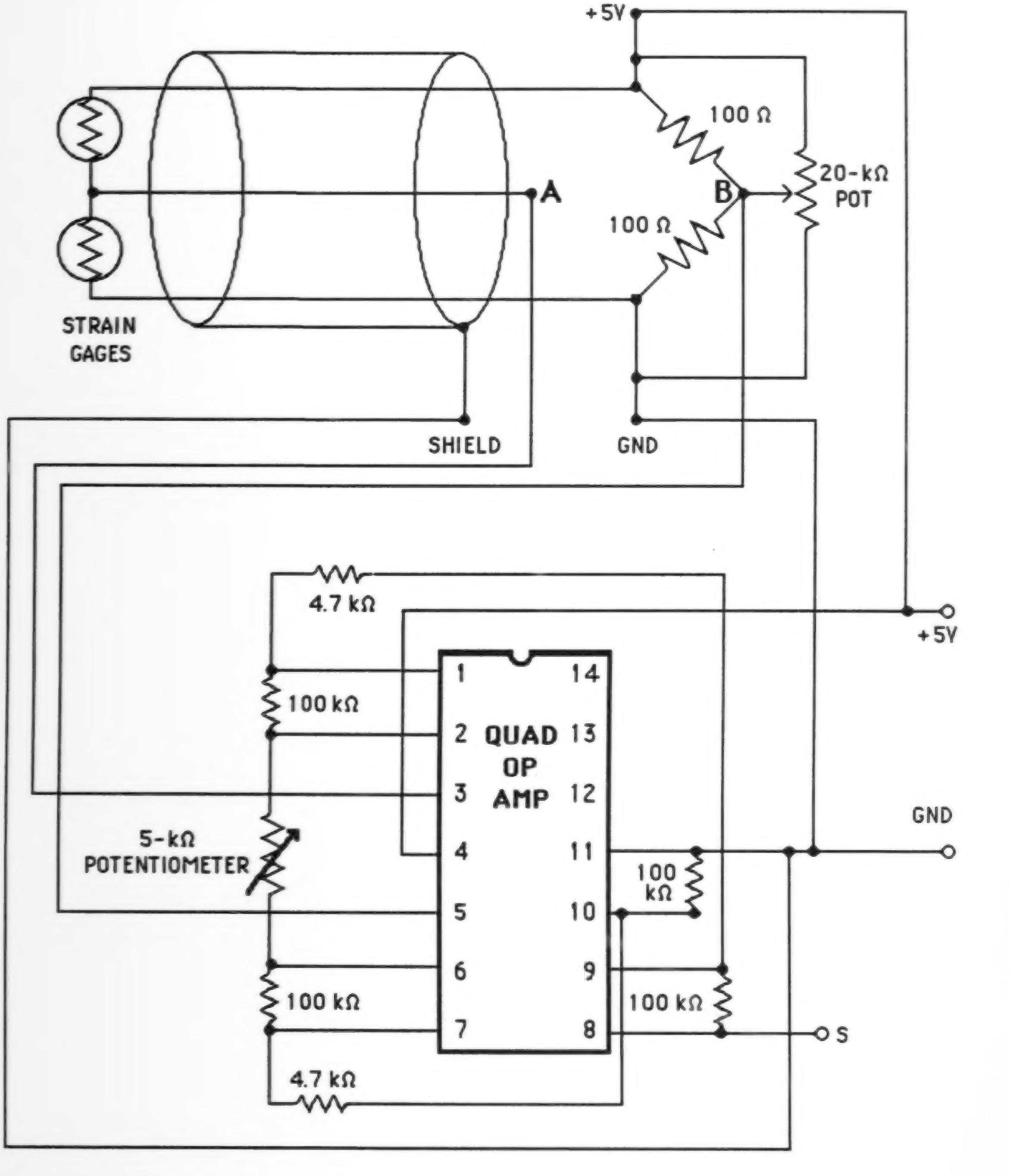
---

[13] If standard (5%) resistors are used, the various resistor pairs will not match up as well. When the resistor pairs match exactly, extraneous electrical noise is most efficiently reduced.

[14] As you study the diagram, you may wonder why we use a separate wire from the SHIELD terminal to the GND lead of the Voltage Monitor (instead of just connecting the SHIELD and GND leads near the bridge). This is done to minimize "ground loop" interference effects.

[15] If necessary, use an ohmmeter to identify which wire connects to which terminal.

[16] Two other projects in this book that require a similar cord are *Thermocouple* (Project 8) and *pH Meter* (Project 9).


+5V
100 Ω
20-kΩ POT
A
B
100 Ω
STRAIN GAGES
SHIELD
GND
4.7 kΩ
+5V
1
14
100 kΩ
2
QUAD OP AMP
13
3
12
GND
5-kΩ POTENTIOMETER
4
11
100 kΩ
5
10
6
9
100 kΩ
100 kΩ
7
8
S
4.7 kΩ

WORKING DIAGRAM OF THE STAIN-GAGE BRIDGE/AMPLIFIER CIRCUIT (TOP VIEW)

☐ 6. **Test the completed circuit.** Turn off the computer. Adjust the gain (5 kΩ) potentiometer to its middle position. Plug the strain gage bridge/amplifier into the Voltage Monitor. Turn on the computer and run the VOLTAGE program on the *Project Programs* disk. Select mode "V". The computer should begin displaying the voltage reading. Adjust the bridge potentiometer (20 kΩ) until the computer reads about 0.5 volts when the metal bar is unstressed. Bend the bar slightly and watch for a change in the voltage reading. The voltage should increase as the force increases. If the voltage decreases when the force increases, either turn the bar upside-down or switch the two leads from the strain gages to points A and B. Adjust the gain control on the amplifier (5-kΩ potentiometer) so the voltage out is about 3 volts when the bar is flexed by the largest force you plan to apply to it. You may need to adjust each of these potentiometers a few times until you get both the unstressed and the full scale readings at the proper settings:

- Unstressed: about 0.5 volts
- With maximum stress: about 3 volts

If you get no change in the voltage when the bar is bent, or if the readings are very unstable, refer to the "Troubleshooting" section below.

☐ 7. **Calibrate your strain-gage system.** With the hardware connected to the computer, run the program STRAIN. Allow the system to "warm up" for a few minutes.[17] Select the calibration mode. With no stress applied to the bar, press <RETURN>. Now place a known force on the bar. This can most easily be done by setting an object of known weight on the bar.[18] Allow a few seconds for the bar to reach equilibrium with the force applied to it, then press <RETURN> again. This time, you will be asked to enter the force applied and the units of force. You can use any standard force units (newtons, pounds, etc.), or you can define your own force units such as "pennies" or "bricks."[19] Press <RETURN> again and the calibration data you entered will be saved on the disk for later use. If you need more information on how to use the STRAIN program, refer to the "Program Design Notes" below.

☐ 8. **Test the system over a range of forces.** Apply several different known forces to the strain-gage bar. This can most easily be done by setting objects of known weight on the bar. Record data and make a graph of force displayed by the computer versus actual force applied.

- Is the graph of the force readings from the system versus actual force a straight line, as it should be?
- What is the maximum error?
- What is the precision of this force measurement system?
- Are the force readings reproducible? If you weigh the same object several times, does the system display the same force?
- Is there a long term drift in the force readings when a steady force is applied?
- Is your system sensitive to electromagnetic fields? Does the reading change when you move your hand near the bar or the amplifier circuit?

TROUBLESHOOTING

The following tests are recommended to help you spot problems. All voltages are relative to ground (GND) unless otherwise stated. All voltage measurements should be made with the system connected and the computer on.

---

[17] The strain gages have a small current running through them. This current produces some heat and slightly raises the temperature of the gages. This in turn changes their resistance. This "self heating" can affect calibration. If you wait a few minutes, the gages and the bar will reach an equilibrium temperature and the reading will stabilize.

[18] The position along the bar where the weight is placed is important. The force must be applied at the same point for all force measurements made using a particular calibration.

[19] A quick way to test the strain-gage system is to calibrate using a number of similar objects, for example, pennies. Then check the system by using it to weigh different stacks of pennies to see if it reads the weight correctly. This test assumes that the objects you are using each weigh the same amount.

If your system always reads a constant voltage (or zero), even when the bar is bent:

◊ BRIDGE PROBLEMS: Use a voltmeter to check the voltage at points A and B. They should both be half of the excitation voltage or 2.5 volts. If the voltages are zero or 5 volts, either you do not have the excitation voltage hooked up or one of the resistors in the bridge is shorted. If the voltage readings at A and B differ greatly, try adjusting the bridge potentiometer (20 kΩ). Remember that the adjusting screw may need to be turned several complete rotations. If that does not allow you to match the voltages, double-check the bridge circuit. Check especially to see if the leads on the strain gage are touching each other or if they are shorted to the metal bar. An ohmmeter can also be used to check the bridge. With the power off, check its total resistance, top to bottom. It should be approximately equal to the resistance of one strain gage (120 Ω). Break the bridge apart temporarily and check each of the four parts of the bridge. All four parts should have about the same resistance.

If the bridge passes the tests above, but you want to make sure the bridge is working properly, hook up a good voltmeter across A and B. Connect the hardware to the computer and turn the computer on. Set the meter to read fractions of a millivolt. Bend the bar. You should be able to detect a change in voltage between points A and B.

◊ AMPLIFIER PROBLEMS: If the bridge seems to working correctly, but you still do not get a change in voltage as the bar bends, check the following:

- Is your IC inserted properly? Pin #1 is marked with a dot. Is it where it should be?
- Have you numbered the pins of the quad op amp properly? Remember that the diagram of the IC is a top view. The numbering looks different when seen from the bottom.
- Does your quad op amp get hot when you turn things on? Hook things up and turn off the computer. Touch the top of the op amp. If it is hot, you must have a serious wiring problem and you may even need to replace the chip after you fix the wiring error.
- Make sure you are using the correct resistors. *Appendix B* gives color codes to help in identifying the resistors.
- Visually double-check all of the connections on the quad op amp. There are a lot of wires and it is easy to have one in the wrong place.

If none of these problems are found, it is time to check the voltages on the various pins of the op-amp IC with a voltmeter. The negative lead of the voltmeter should be connected to the GND line from the Voltage Monitor.

PIN #	VOLTAGE SHOULD BE	IF NOT, DOUBLE-CHECK
4	+5 V	Power supply lead from Voltage Monitor
11	0 V	Wrong wire connected to pin #11
3 and 5	2.5 V	Problem with bridge or wrong wire connected to these pins
8	0 to 3 V	This is the output voltage and it should change as the bar is bent

If the voltage at pin #8 does change (as it should) as the bar bends, then the problem must be in the Voltage Monitor or its connections. Check the Voltage Monitor by applying a known voltage signal.

If the system is slow to respond:

This is probably a problem with the physical setup of the strain gages on your bar. The strain gages are not moving with the bar as they should. Perhaps you have not glued them firmly and evenly to the bar; or your glue may be too thick. If there is a significant layer of glue between the strain gage and the bar, the gage will gradually "flow" into a new position. You can either live with this problem of slow response or glue on a new pair of strain gages. You may want to consider using a special strain-gage adhesive.

If the readings fluctuate widely:

If the output of the strain gage bridge-amplifier fluctuates considerably, the most likely problem is stray signals that are picked up on the bridge and then amplified. To minimize this:

- Make sure the cable between the strain gages and the circuit board has a metal shield that is connected to the GND lead from the Voltage Monitor.
- Keep the cable between the strain gages and the circuit board as short as possible.
- Make sure all connections inside the bridge-amplifier are neat and short. Any extra loops of wire act as antennae for stray signals.
- Place the bridge-amplifier circuit inside a grounded metal box (see the "Project Extensions").

PROJECT EXTENSIONS

Circuit Modifications

Note: If you are interested in really "fine tuning" your strain-gage system for more accurate readings, you should request the literature from the commercial strain-gage manufacturers mentioned at the end of the "Background Information" section above.

◊ Place the bridge/amplifier circuit inside a metal box

A metal box can help shield out electromagnetic fields. Connect the box to the GND lead from the Voltage Monitor.

◊ Add a voltage regulator to the circuit

The accuracy of the strain-gage, force-measurement system depends on the excitation voltage that is applied to the bridge circuit remaining constant. In the circuits above, we use the +5V line from the Apple II power supply for the excitation voltage. The voltage of this line is fairly constant, but you can guarantee a constant excitation voltage by using a precision voltage regulator. Several adjustable voltage regulators are sold by Radio Shack stores (RS #276-1778, RS #276-1740).[20]

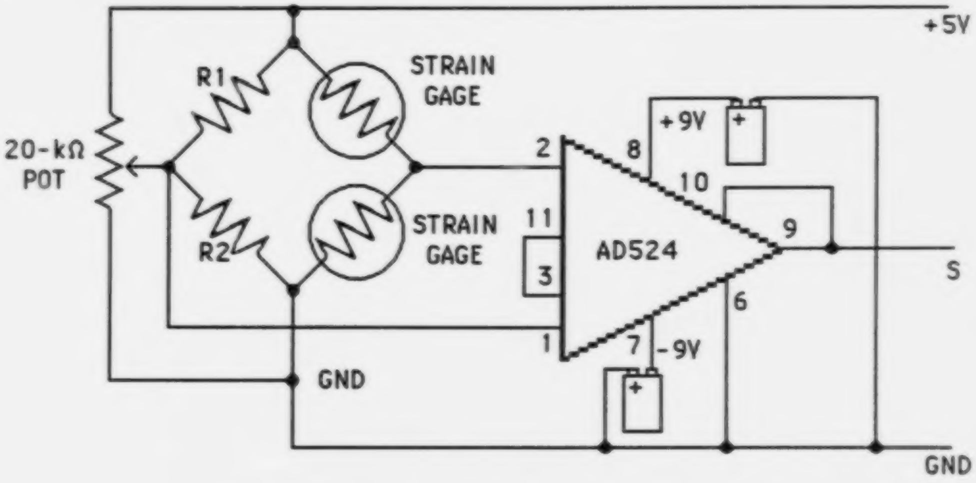
◊ Use a commercial instrumentation amplifier

There are several integrated circuit manufacturers who make ready-to-use instrumentation amplifiers. One example is the Analog Devices AD524.[21] It is essentially an improved, assembled version of the instrumentation amplifier that we describe in the "Step-by-Step Instructions." It has *programmable gain*, which means you select the amplification factor by how you connect one of its pins. The circuit below shows the total circuit needed for this project using an AD524. Notice that the circuit is much simpler than the core project circuit, since the whole instrumentation amplifier is built into the one IC. This circuit will give you more stable, reliable results. Also, note that the AD524 (like most commercial instrumentation amplifiers) requires a dual power supply. This means you need to use two 9-volt batteries in the circuit. As wired below, the gain is 1000.

---

[20] There is a second advantage to using a voltage regulator. Since you set the output voltage of the regulator at a voltage less than 5 volts, the "self heating" of the strain gages is reduced.

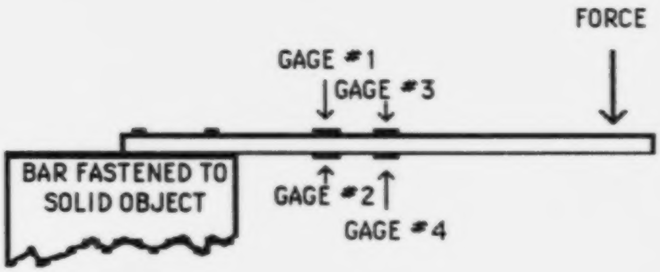
[21] The AD524 is sold by Analog Devices Inc., Two Technology Way, P.O. Box 280, Norwood, MA 02062. The AD524 comes in various grades, indicated by the last two letters in the part number. We usually use the AD524BD, which is a medium grade version that sells for approximately $20.00.

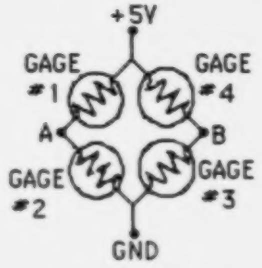

+5V
R1
STRAIN GAGE
20-kΩ POT
R2
STRAIN GAGE
+9V
2
8
10
11
AD524
9
3
S
6
1
7
-9V
GND
GND

COMMERCIAL INSTRUMENTATION AMPLIFIER CIRCUIT

◊ Build a strain-gage bridge using four active gages

The sensitivity of the force-measurement system can be improved if four strain gages are mounted on the bar. The voltage difference produced by a four-gage bridge is twice as much (for the same amount of bending) as with a two-gage system. The four gages should be wired as shown below:

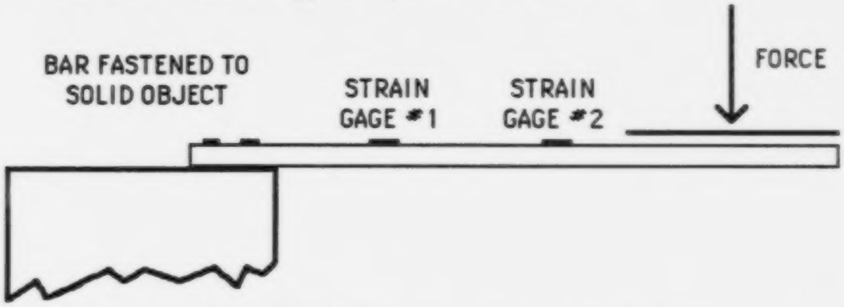

FORCE
GAGE #1
GAGE #3
BAR FASTENED TO SOLID OBJECT
GAGE #2
GAGE #4


+5V
GAGE #1
GAGE #4
A
B
GAGE #2
GAGE #3
GND

4-GAGE BRIDGE

◊ Build a differential strain-gage system

If the two strain gages used are mounted as shown below, the sensitivity of the system is reduced, but the point at which the force is applied becomes less critical.


FORCE
BAR FASTENED TO SOLID OBJECT
STRAIN GAGE #1
STRAIN GAGE #2

DIFFERENTIAL STRAIN-GAGE SYSTEM

With a standard strain-gage setup, the force must always be applied at the same spot for accurate readings. With a differential strain-gage system, the point at which the force is applied makes no difference, as long as it is between strain gage #2 and the end of the bar. This is a good arrangement for many projects, such as building a "scale" to weigh people. With this bridge design, the exact positioning of the person's feet will not affect the weight displayed.

Program Improvements

The STRAIN program on the *Project Programs* disk is a fairly simple program that can be improved by adding new features. The "Program Design Notes" section below explains how the STRAIN program operates. It should help you in making modifications. Some suggested changes are given below:

- Allow the force and time data to be saved on a disk and reloaded later.
- If your computer has a "clock card" to keep track of real time, change the program so it reads the clock time and stores it along with the force reading.
- Program the computer to record the maximum and minimum force readings encountered.
- Add a mode to the program which graphs the force readings versus time.
- Add a mode to the program which displays the force in large low-resolution graphics block digits so they can be read at a distance.

Additional Experiments

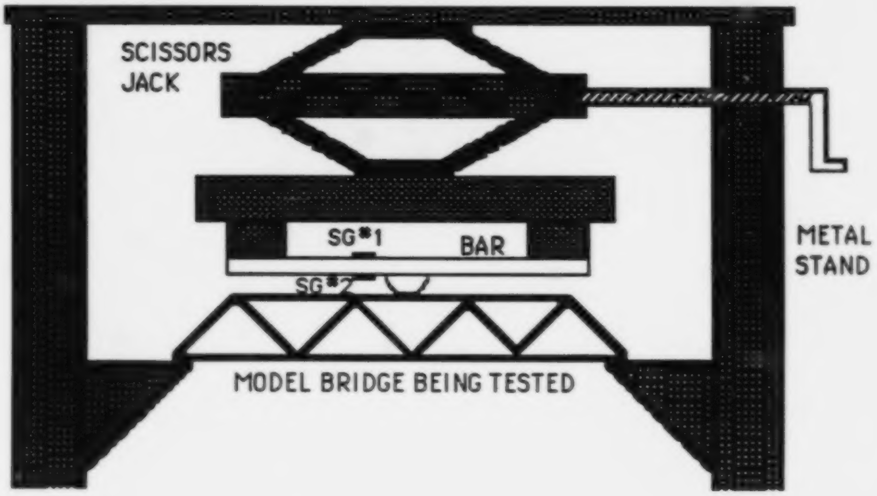
The completed force-measurement system can be used for many interesting projects. A few examples are listed below:

- Study drag forces in a wind tunnel

  Construct a wind tunnel using an electric fan. Mount objects on the end of a strain-gage bar so the force of the wind pushing back on the object (drag) can be measured. In a similar way, "lift" forces can be measured using various shapes of wings.

- Model bridge testing

  Many science classes hold contests to see who can build the strongest model bridge using a limited amount of materials (typically balsa wood or toothpicks). Testing the bridges is usually done by hanging weights from the bridge until the bridge breaks. A better alternative is to apply force to the bridge using a jack mounted so that it pushes down on the bridge. One arrangement for doing this is shown below:



◊ Fat tank

The most accurate way of determining a person's percentage of body fat is to "weigh them underwater." The person sits in a seat suspended from a rope. The rope is attached to a force measurement system (a strain-gage bar perhaps). They are weighed first in the air. Then they are lowered completely underwater.[22] Their "weight" changes due to the buoyancy of the water. Their percentage of body fat can be determined from the two different weight readings.[23]

SUPPLEMENTAL INFORMATION

Program Design Notes

The STRAIN program on the disk will do the calculations necessary to convert the voltage readings from the strain-gage amplifier to force measurements. Here is a section-by-section explanation of the program.

Lines 100 to 990: Initialization
This section loads the machine language FREQ program. It then reads the STRAIN.CAL file, which gives the program the conversion constants needed to calculate force from voltage. It also displays the main menu.

Lines 1000 to 2990: Display Force
This section simply reads the force and displays it on the monitor.

Lines 2000 to 2990: Display Force and Save in Memory
This section displays the force readings and also saves them in memory. Up to 500 force readings can be saved in memory. When this mode is selected, you will be asked:

```
ENTER DELAY TIME (TIME BETWEEN FORCE
READINGS STORED IN MEMORY) [1]
```

This option will allow you to save force measurements made over a long time period. This is useful in a study of the stability of the strain-gage system. If you enter a time delay of 5, for example, the reading of the strain gage will be checked every second, but the readings will be saved in memory only every 5 seconds. In most cases, you do not need this delay. If you simply press <RETURN> or enter a 1, the readings will be saved in memory as fast as they are made, which is once a second.

After you enter this value, the program will begin monitoring the force. Any time the force reading is saved in memory, the line on the data table is marked "<- SAVED". The screen will display both the time (in seconds) and the force. The times measured by the program are only approximate.

Lines 3000 to 3990: Calibration
This section allows you to calibrate the strain gage. First, with no stress on the bar, press the<RETURN> key. Then, add a known weight (or a known, steady force). Let the reading stabilize and press <RETURN> again. The program will then ask you to enter the force used and the force units (newtons, pounds, coins, etc.). When you enter these quantities, the program will save the calibration data on the disk in the file STRAIN.CAL. This data will then be used each time the program starts up.

Lines 4000 to 4990: Table of Data
This section lets you display a table of data collected using the "S = SAVE FORCE DATA IN MEMORY" option. Both the force and the time at which the force was measured are displayed.

---

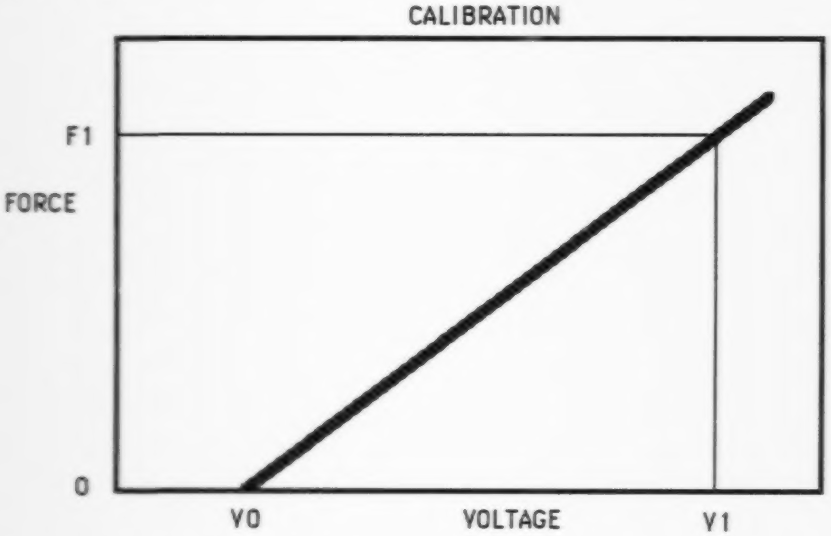
[22] This underwater procedure should be completed within a few minutes, or the person will not care what their percentage of body fat is.

[23] One reference on this topic is "It's All Muscle" in *Health*, October 1983, p. 6.

Lines 10000 to 10990: Voltage Check Subroutine
These lines are used to do the check of the voltage from the Voltage Monitor. Refer to the "Program Design Notes" for Project 7 for details.

How the Voltage is Converted to a Force

The graph below shows how force is calculated. The output voltage of the strain gage is assumed to be proportional to force. The "zero force" voltage reading is determined during calibration. F1 is the force reading when the known force is used during calibration. V1 is the voltage reading from the strain gage with that force on it.



The slope of the line between these two points is calculated from:

slope = F1 / (V1-V0)

The slope, the value of V0 and the force units are saved in the STRAIN.CAL file.

To convert a voltage reading (V) to a force value, the program picks the value along the line on the graph by calculating:

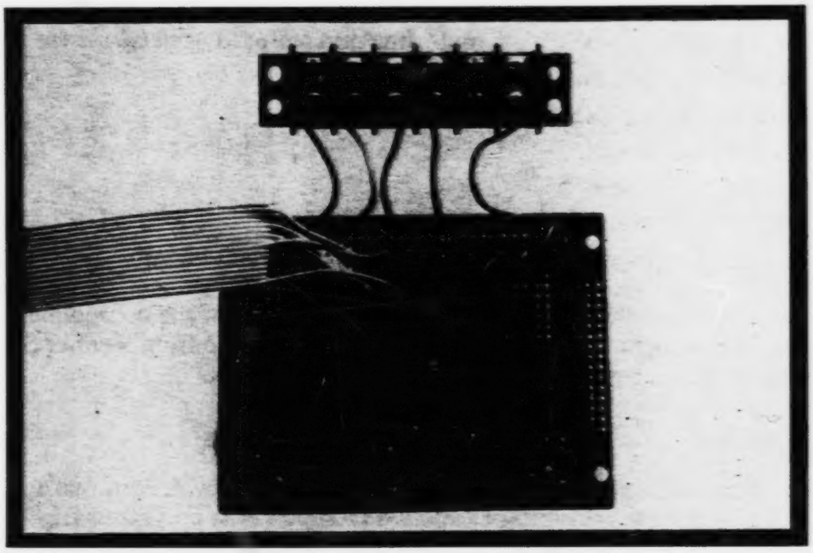
force = slope (V - V0)

Note that this program assumes that the voltage out of the bridge/amplifier is proportional to the force applied to the bar. You should check this assumption as part of your testing.

The following variables are used in the STRAIN program:

C	Counts from the Voltage Monitor	Q$	Dummy input string
D$	Control D, used for disk access	SL	Slope
F	Current force reading	T	Time (in seconds)
F1	Force used in calibration	TD	Time delay for saving data
F( )	Force data saved in memory	T( )	Time data saved in memory
I	Delay loop counter	U$	Force units
J	Data line counter	V	Present voltage reading
M$	Mode of operation	V0	Voltage reading with no stress on bar
N	Number of data lines	V1	Voltage reading with known force on bar

PROJECT 11:
OPTICALLY-ISOLATED SWITCHES



This project consists of building four on/off switches that can be controlled from the Apple game port. The switches are *optically isolated* from the computer. This means that there is no electrical connection between the switches and the computer. Optical isolation virtually eliminates the possibility of damaging the computer as you work with external electrical circuits. The switches can be used to turn on and off LEDs, light bulbs, motors and other direct-current (DC) devices. This project will also be used as a starting point for the three projects that follow: *Computer-Controlled Car*, *Stepper Motor* and *A Better Mousetrap*.

PROJECT REQUIREMENTS

Background Knowledge

- Electronics: In terms of the complexity of the circuit, this is one of the most difficult projects in this book. Also, the "Step-by-Step Instructions" are not as specific as on most of the projects. More of the details of component placement are left to you. Previous experience in assembling electronic circuits (for example, other projects in this book) is recommended.
- Programming: No programming experience is required for the core project. Some of the extensions require experience with BASIC programming.

Parts

☐ (4) optoisolators, also known as optical isolators or optocouplers (Almost any optoisolator with NPN phototransistor output or NPN photodarlington output will work. Do **not** use an optoisolator with "triac output" [Radio Shack sells one]. Here is a partial list of specific part numbers we have used successfully: PS2002B [Mouser Electronics #551–PS2002], NTE3040, NTE3045, 4N28, H11A1 and TIL111. Radio Shack sometimes sells an inexpensive package of mixed optoisolators [RS #276-139]. The darlington-output and transistor-output optoisolators in this package will probably work.)

☐ 7404 Hex Inverter Integrated Circuit (RS #276-1802)

☐ 14-pin IC socket (RS #276-1999)

☐(4) 8-pin IC sockets (RS #276-1995)
☐(4) MPS2907, PNP transistors (RS #276-2023)
☐(4) TIP3055, NPN transistors (RS #276-2020)
☐(4) 1-amp diodes, PIV 50 volts or more (RS #276-1101 is a package of two)
☐ 16-connector cable to Apple's internal game port (refer to *Appendix A* for information)
☐ large circuit board (one suggestion is RS #276-162)
☐ 6 position barrier strip (RS #274-659)
☐(4) 330-Ω resistors, 1/4 watt, 5%
☐(4) 47-kΩ resistors, 1/4 watt, 5%
☐(4) 1-kΩ resistors, 1/4 watt, 5%
☐ miscellaneous jumper wires for connections on the circuit board

Note: An "Optically-Isolated Switches Parts Kit" containing all of the components needed for this project, including a 16-conductor cable to the game port, is available from Vernier Software. This kit sells for $30.00. An order form is provided at the back of this book.

Tools and Supplies

Solder, soldering iron, multimeter, pliers, drill, drill bits, screwdriver, wire cutters and wire strippers. You will also need a low-voltage (3 to 9 VDC) power supply, four LEDs and four 220-Ω resistors for testing the circuit.

BACKGROUND INFORMATION

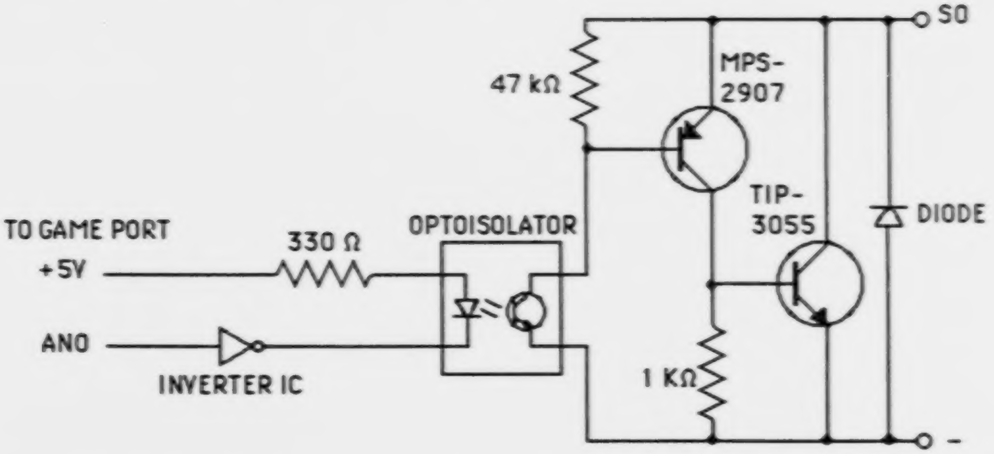
One interesting way to use a computer is for controlling various electrical devices. Fortunately, the Apple II, II+ and IIe game ports have four connections that can easily be used for this purpose. These terminals are called *annunciator outputs*. These terminals are labeled "AN0", "AN1", "AN2" and "AN3" on the 16-pin internal game port. These terminals are not available on the 9-pin external game port. Each of these annunciator lines can be controlled with BASIC statements. If you have not already done so, read "The Apple Game Port" section in the *Introduction* for information on how the annunciator lines are controlled.

The annunciator-output lines are easy to control, and it would be nice if you could simply hook them directly to electrical devices . Unfortunately, it is not that easy because the lines cannot handle very much current.[1] To control useful devices with these lines, you need to use an electronic circuit as a "buffer" between the computer and the outside world.

The circuit we use in this project will allow the computer to switch on or off four different direct-current (DC) electrical devices. Each of the devices can use as much as 10 amperes of current. The circuit can be used with DC voltages up to 25 volts. A complete circuit for a single switch is diagrammed below.

---

[1] The annunciator lines are designed to drive only one "TTL" load. Refer to "The Apple Game Port" section of the *Introduction* for more information on the annunciator lines.



SCHEMATIC DIAGRAM OF A SINGLE OPTICALLY-ISOLATED SWITCH

We will look at the circuit part by part. The annunciator line from the Apple game port is connected to a ***7404 Inverter***. This integrated circuit is used because it can safely "sink" more current than the Apple annunciator lines.[2] The IC is called an inverter because it changes the input signal it receives from low to high or vice versa. For example, if the AN0 line is at its high-voltage level, the output of the 7404 inverter (right side in the diagram above) will be a low voltage.

A key part of the circuit is the **optoisolator**. Optoisolators are small devices that look like integrated circuits. They include an LED (Light-Emitting Diode) and a phototransistor. In the circuit above, whenever the output of the 7404 is in its low state, current flows from the Apple's +5V line through the 330-Ω resistor and through the LED inside the optoisolator. The light produced by the LED affects the phototransistor inside the optoisolator. The phototransistor acts like a switch that is "closed" (conducting) whenever light strikes it, and "open" (not conducting) when no light is striking it. Since the phototransistor is sealed inside the case of the optoisolator, it can receive light only from the LED.[3] There is no electrical connection between the two sides of the optoisolator. The only "connection" is by light.

When the phototransistor inside the optoisolator switches on, current flows through the 47-kΩ resistor, lowering the voltage at the base of the **MPS-2907 transistor**. This transistor is now "turned on." Current can now flow through it and the 1-kΩ resistor. This raises the voltage at the base of the **TIP-3055 transistor**. When the voltage of the base of the TIP-3055 rises above the level of the "-" terminal, it is switched on. Current can now flow from the S0 terminal, through the TIP-3055 transistor, and to the "-" terminal. The Optically-Isolated Switch is then "closed."[4]

The **diode** in the circuit is used to protect the transistors from electrical damage when the switch is used to control *inductive loads*. An inductive load is any device using an electromagnet, such as a motor or a relay. Whenever the magnetic field of an electromagnet is turned off, the energy of that field causes a short pulse of current flowing backward in the electromagnet circuit. The diode in the switch circuit provides a path for this current around the transistors of the switch. Without the diode, the transistors can be damaged by the reverse current. The diode also protects the switch circuit in case an external power supply is ever connected to the circuit with the "+" and "-" terminals reversed.

---

[2] The 7404 Inverter is a standard TTL device. Its outputs can safely sink a current of 16 mA.

[3] These are the same components used in the photogate from Project 1. An optoisolator is similar to a miniature photogate enclosed in an IC package.

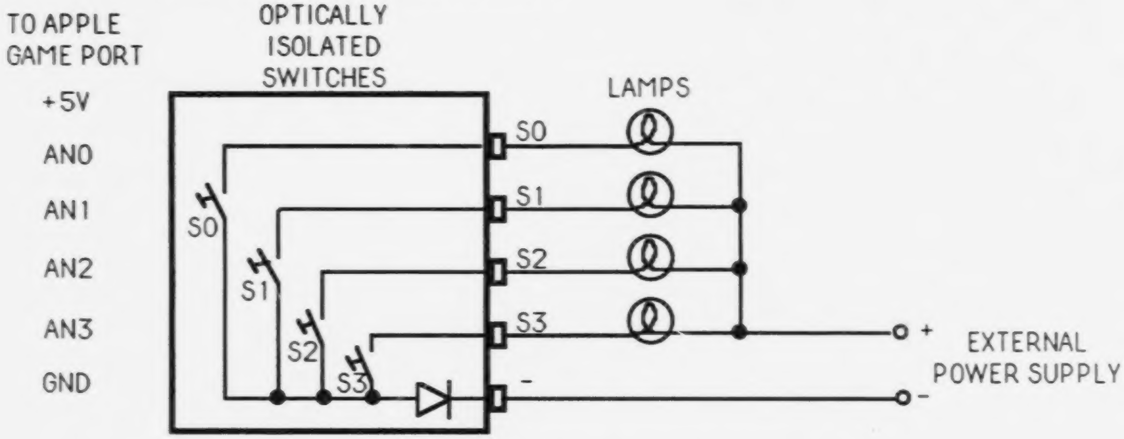
[4] For a complete discussion on how transistors operate as switches, see *Interfacing & Digital Experiments with your Apple* by Charles J. Engelsher.

The circuit diagrammed and discussed above is for a single DC on/off switch. A similar circuit can be used with each of the four annunciator-output lines. The "Step-by-Step Instructions" explain how to build a circuit to control four switches.[5]

This may seem like a lot of trouble to produce a computer-controlled switch. It is true that there are easier ways to get the job done; however, this system has a number of advantages:

- There is no electrical connection between the computer and the external circuit. This protects the computer from any accidental electrical damage. Even if you inadvertently touched a high-voltage lead to the right side of the circuit, it is very unlikely that the computer would be damaged. There is no electrical connection at all between the phototransistor side of the circuit and the computer.[6]
- The current sunk into the annunciator line and the current drawn from the game-port +5V power line are within the recommended limits.[7]
- The switch can be used with inductive loads: electromagnets, motors, relays, etc.
- The switch can safely handle large currents (up to two amperes, and more if an extra heat sink is added).[8]
- The switch can operate with DC voltages from 0 to 25 volts.
- The circuit is protected against damage if the external power supply is connected backward.
- Each of the four annunciator lines can be used to control a different switch.

The finished Optically-Isolated Switches can be used for controlling many different circuits. Five terminals on a barrier strip can be used to connect the various circuits to be controlled. A typical circuit, used to control four lamps is shown below:



FUNCTIONAL DIAGRAM OF THE OPTICALLY-ISOLATED SWITCHES

The part of the circuit inside the rectangle is intended to represent how the Optically-Isolated Switches function. When one of the four annunciator output lines (AN0, AN1, AN2 or AN3) is turned "on," the corresponding switch terminal S0, S1, S2 or S3 is electrically connected to the "-" terminal. This completes an electrical circuit, allowing electricity to flow from the positive terminal of the external power supply through the lamp, then through the switch, and back to the power supply.[9] The *Project Programs* disk includes programs to turn on and off the four switches from the computer keyboard or from within a program. You may want to write your own programs to control the switches.

---

[5] For some purposes, you may want to build just one or two optically-isolated switch circuits.

[6] Most optoisolators have an *isolation voltage rating* of several hundred or even several thousand volts. If a potential difference greater than the isolation voltage rating occurs between the two parts of the optoisolator, the electrical separation may break down.

[7] Apple Computer, Inc. recommends you avoid drawing more than 100 mA from the Apple's +5V line.

[8] The addition of the heat sink on the TIP3055 transistor is left as one of the project extensions.

The next three projects in this book involve using the Optically-Isolated Switches to control DC motors and stepper motors. Several other uses for the switches are suggested in the "Project Extensions."

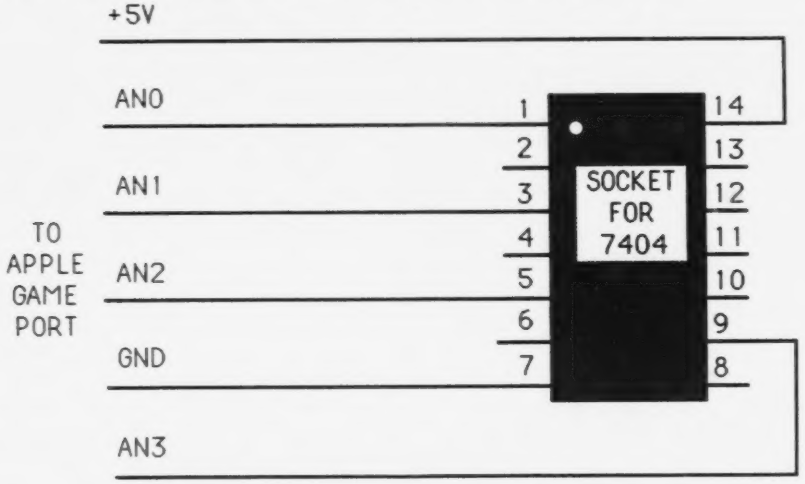
STEP-BY-STEP INSTRUCTIONS

☐ 1. **Experiment with component placement.** Before you solder anything, take a few minutes to experiment with the placement of the major components onto the circuit board. An efficient layout can save a lot of wiring and time. Also, you do not want to get halfway through the soldering and find there is not enough room on the circuit board you have chosen. Basically, you need to have room for one 14-pin socket with not too much around it and four 8-pin sockets (for the optoisolators). Near each optoisolator on the pin #1 to #3 side will be one resistor. The other side of each optoisolator needs room for two resistors, two transistors and a diode. Refer to the following working diagrams (especially in steps 5 and 6) to get an idea of the way the components will need to be connected.

☐ 2. **Solder the 14-pin socket to the circuit board**. Place the 14-pin socket (for the 7404) on the circuit board and solder all of its pins. Note which of the corners of the socket is designated for pin #1 of the 7404.

☐ 3. **Connect the 16-conductor game-port cable.** The AN0, AN1, AN2, AN3, GND, and +5V leads will be used in this project. Refer to *Appendix A* for information on identifying these leads. Solder the connections for these six leads as shown in the diagram below.

Note: For some of the other projects using this circuit, you may want to make connections to the PB2, PDL0 or PDL1 lines on this ribbon cable. Keep these leads accessible.

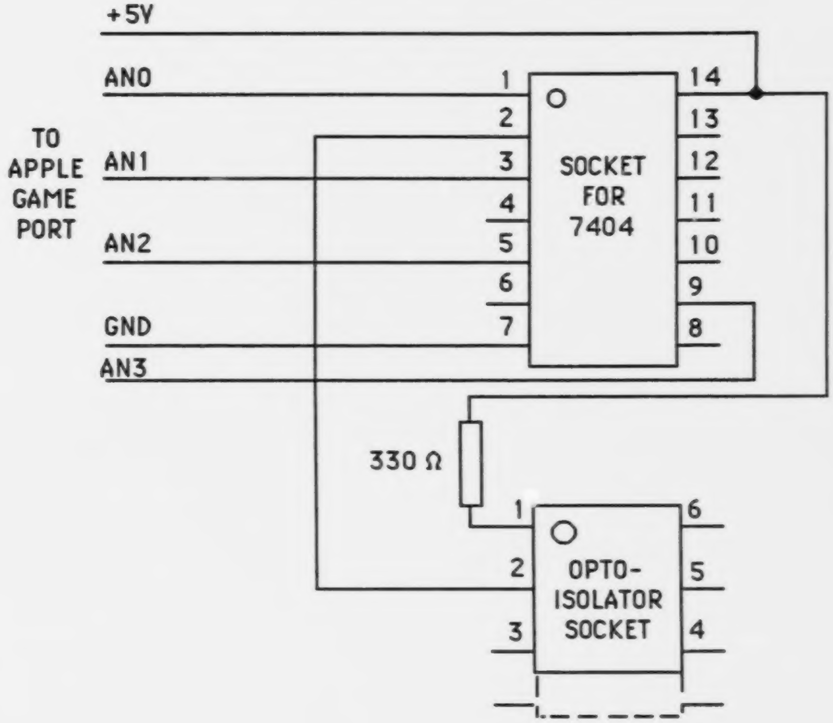


---

[9] This is a diagram indicating what the switches do, not a schematic diagram of the circuit. A diode is included in the diagram for two reasons: (1) It represents a voltage drop of 0.7 volts or more that always occurs across the switches when they are "on;" and (2) It is a reminder that these switches cannot be used with alternating current (AC).

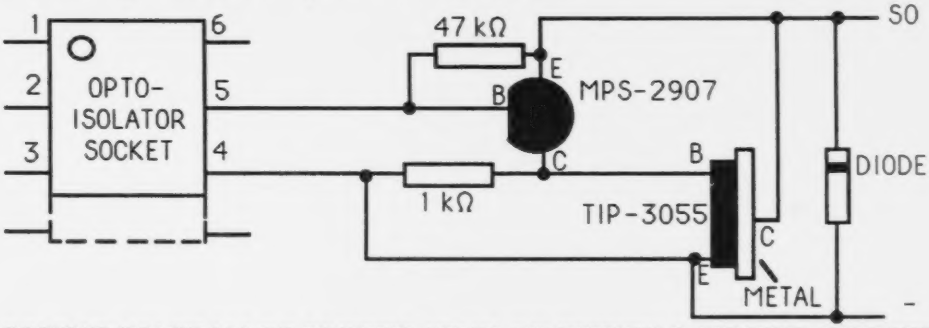
☐ 4. **Add an 8-pin socket to the circuit board for one optoisolator.** Place an 8-pin socket on the circuit board for use with the switch 0 optoisolator. The optoisolator will later be plugged into the socket, leaving holes 4 and 5 empty.[10] Solder the socket in place.

☐ 5. **Add jumper wire(s) and the 330-Ω resistor to complete the "computer side" of the circuit for switch 0.** The connections are shown in the diagram below. One wire extends from pin #2 of the 7404 to pin #2 of the optoisolator. Next, connect the 330-Ω resistor between pin #14 of the 7404 and pin #1 of the optoisolator. You may need a jumper wire to complete these connections.


+5V
AN0
TO
APPLE
GAME
PORT
AN1
AN2
GND
AN3
1
2
3
4
5
6
7
14
13
12
11
10
9
8
SOCKET
FOR
7404
330 Ω
1
2
3
6
5
4
OPTO-
ISOLATOR
SOCKET

WORKING DIAGRAM OF "COMPTER SIDE" OF SWITCH 0 CIRCUIT (TOP VIEW)

☐ 6. **Assemble the "power side" of the switch 0 circuit.** The diagram below shows the connections to the right side of the optoisolator for switch 0.


1
2
3
6
5
4
OPTO-
ISOLATOR
SOCKET
47 kΩ
E
B
MPS-2907
C
1 kΩ
B
TIP-3055
C
E
METAL
DIODE
S0
-

WORKING DIAGRAM OF "POWER SIDE" OF SWITCH 0 CIRCUIT (TOP VIEW)

---

[10] You may be wondering why we do not recommend using a 6-pin socket here, since the optoisolator has only 6 pins. 8-pin sockets are the smallest normally sold.

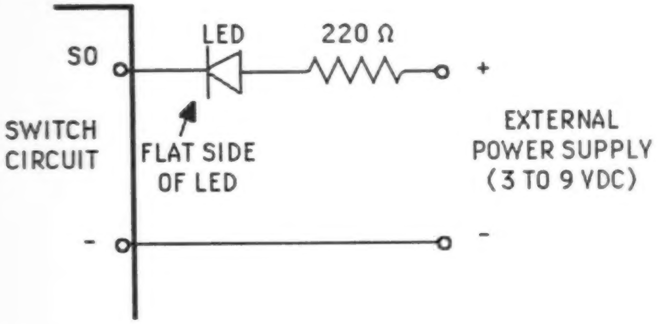
When placing the components on the board for this part of the circuit, be careful about the following:

◊ The pin numbers shown for the optoisolator correspond to the pin numbers of the 6-pin IC, not the 8-pin socket.
◊ Be sure to orient the two transistors properly. The three leads (**C**ollector, **E**mitter and **B**ase) of the transistors are labeled on the diagram above. If you are using a 2907 transistor in a plastic case with one flat side, the diagram can be used to determine the proper orientation for that transistor. If your 2907 transistor is in another type of package, consult the package diagrams to identify the leads. The TIP-3055 transistor should be oriented with the heat sink (metal part) away from the optoisolator. You may need to ream out the holes in the circuit board for the TIP-3055 leads.
◊ Be careful with the orientation of the diode. The end with the band on it must be connected electrically to the middle terminal of the large (TIP-3055) transistor.
◊ Notice that there is no connection to pin #6 of the optoisolator.

Place all of these components on the circuit board. Use short jumper wires to make connections wherever needed. Double-check for the proper orientation of components and wiring connections, and then solder the circuit together. When soldering the transistors, use a heat sink on the leads to prevent overheating.

☐ 7. **Plug the components into the IC sockets**. Plug the 7404 Open-Collector Inverter into the 14-pin socket. Be careful to orient pin #1 properly. Plug the optoisolator into the 8-pin IC socket. Place pin #1 of the optoisolator into hole #1 on the socket. Notice that the bottom two holes in the 8-pin socket are not used.

☐ 8. **Check switch 0**. Make sure this switch is working before you assemble the other three switches. There is no sense in making the same mistake four times. Set up the following simple circuit for testing the switch:



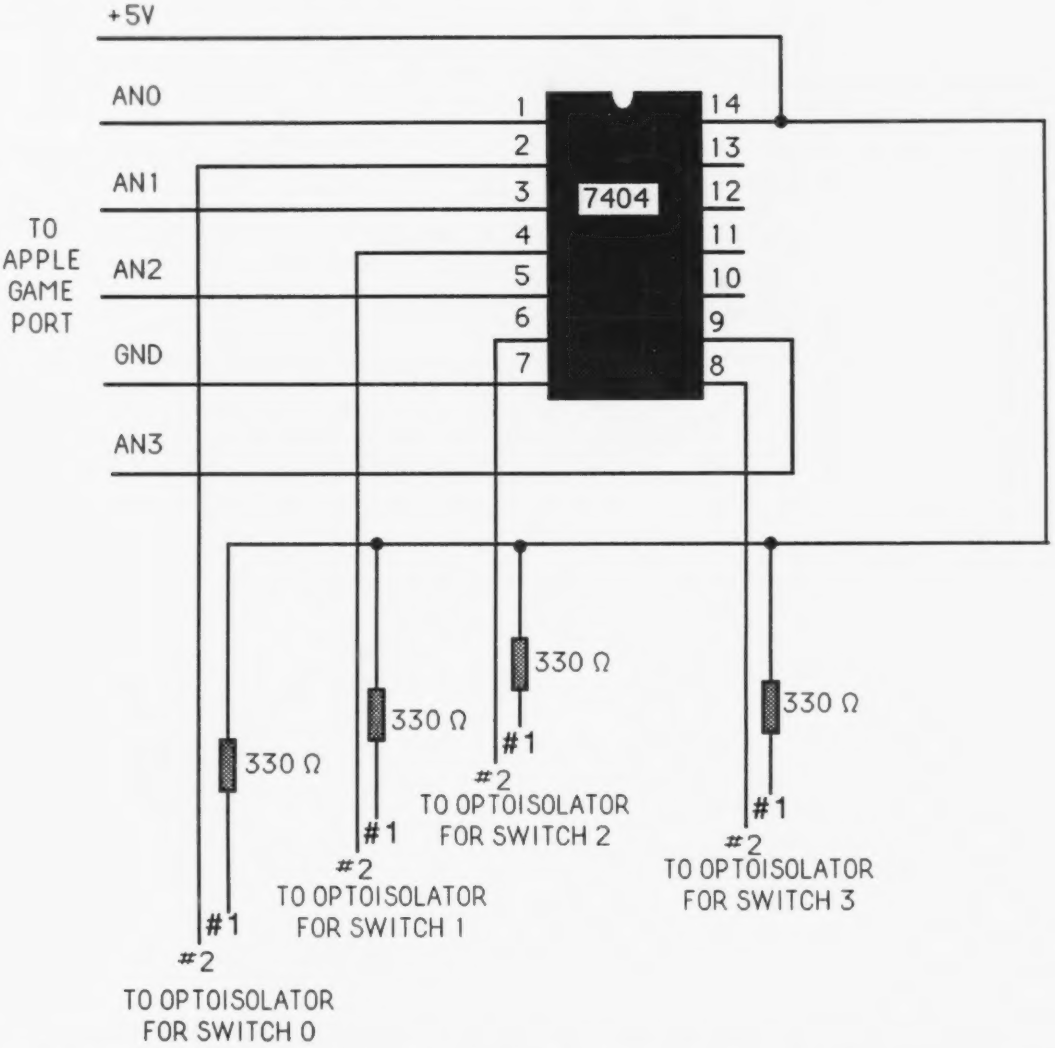
Notice that the flat side of the LED is connected to the switch. You can either temporarily twist or solder the leads together, or use jumper wires with alligator clips on them to make the connections. Instead of LEDs, you can use a small light bulb. If you use a bulb, you do not need the resistor. Make sure you use an external power supply that provides the correct voltage for the bulb.

With the computer turned off, plug the ribbon cable into the 16-pin game socket. Place the *Project Programs* disk in the disk drive and turn on the computer. If the computer does not start up normally, turn it off and check for a mistake in the wiring. Otherwise, make a quick check of the circuit by looking, smelling and feeling for anything hot. None of the components should be warm. If they are, there is a problem with the circuit.

If everything is cool (literally), run the program FLASHER. This program should turn the light source off and on at regular intervals.

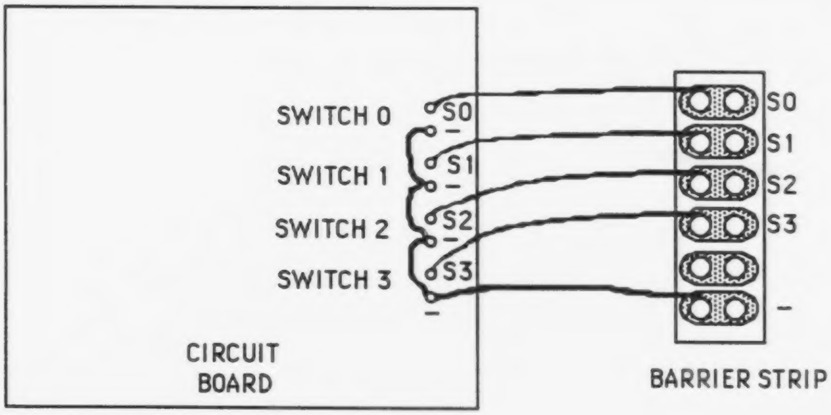
If the light source does not operate as expected, shut off the computer, disconnect the external power supply and double-check the circuit. If you still cannot find the problem, refer to the "Troubleshooting" section below for suggestions on locating the problem.

☐ 9. **Assemble the other three switch circuits.** After you have the switch 0 circuit working, proceed with the other switch circuits. The wiring is the same as the wiring for switch 0, except that different connections from the 7404 IC are used. The connections for all four switches are shown below:

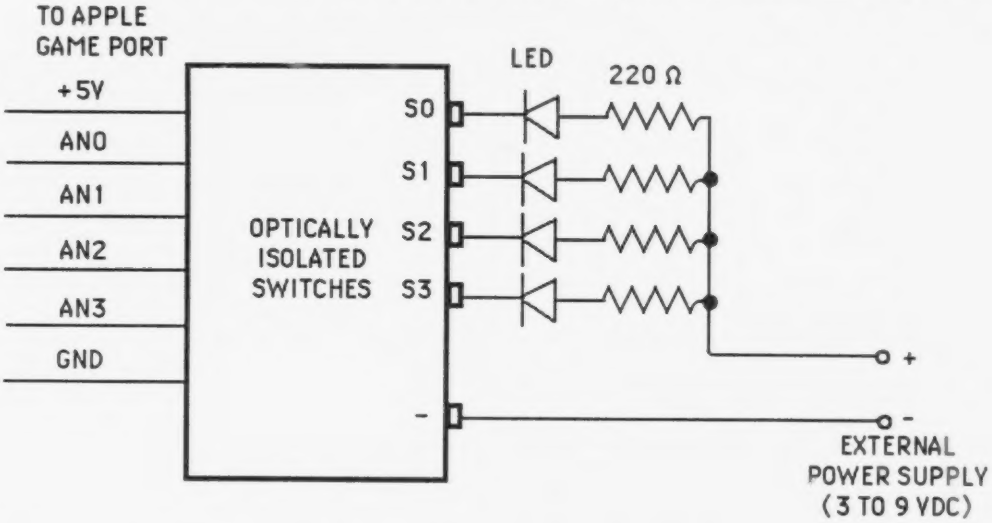

+5V
AN0
AN1
AN2
GND
AN3
TO APPLE GAME PORT
7404
1
2
3
4
5
6
7
14
13
12
11
10
9
8
330 Ω
330 Ω
330 Ω
330 Ω
#1
#2
TO OPTOISOLATOR FOR SWITCH 0
TO OPTOISOLATOR FOR SWITCH 1
TO OPTOISOLATOR FOR SWITCH 2
TO OPTOISOLATOR FOR SWITCH 3

☐ 10. **Plug in the other optoisolators.** Carefully insert the three other optoisolators into the 8–pin IC sockets. Place pin #1 of the optoisolator into hole #1 on the socket. Holes #4 and #5 of these sockets are not used.

☐ 11. **Connect the leads to the barrier strip.** Connect the "-" terminals of all four switches together with jumper wires. Connect a jumper wire from these "-" terminals to the barrier strip. Connect wires from the other side of each of the switches (points S0, S1, S2 and S3) to the barrier strip. A suggested plan for using the five terminals on the barrier strip is shown below:


SWITCH 0
SWITCH 1
SWITCH 2
SWITCH 3
S0
S1
S2
S3
–
CIRCUIT BOARD
S0
S1
S2
S3
–
BARRIER STRIP

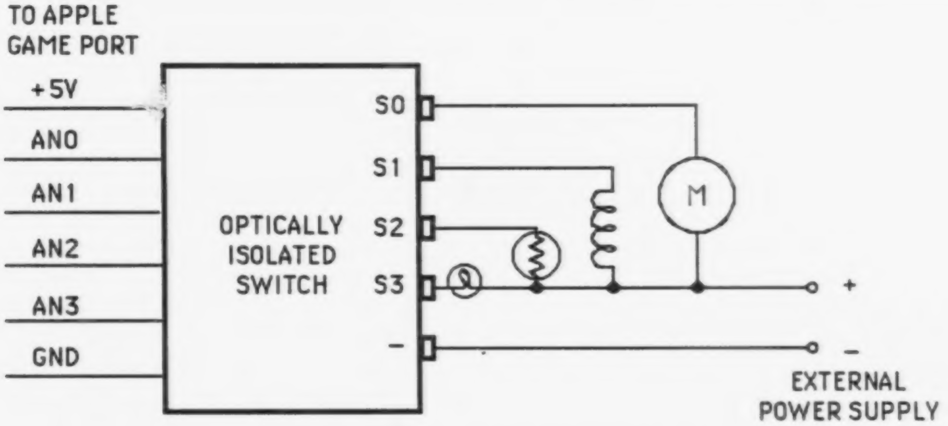
☐ 12. **Test all four switches.** Set up the following simple circuit for testing the switches:


TO APPLE GAME PORT
+5V
AN0
AN1
AN2
AN3
GND
OPTICALLY ISOLATED SWITCHES
S0
S1
S2
S3
-
LED
220 Ω
+
-
EXTERNAL POWER SUPPLY (3 TO 9 VDC)

With the computer turned off, plug the ribbon cable into the 16-pin game socket. Place the *Project Programs* disk in the disk drive and turn on the computer. Check the circuit for any overheating.

Run the program FLASHER. If all is well, this program should turn each of the LEDs on, one at a time.

☐ 13. **Use the Optically-Isolated Switches to control various devices.** Try out the switch with small motors, lamps or radios. The diagram below shows a sample circuit with a motor connected to S0, an electromagnet connected to S1, a heater connected to S2, and a lamp connected to S3.



Any direct current (DC) device that operates on less than 25 volts and draws less than 2 amperes of current can be used. One of the project extensions shows a small addition to the circuit that will allow the switches to operate with currents up to 10 amperes.

Try out the program SWITCH that is on the *Project Programs* disk. This program lets you control each of the four switches individually.

WARNING: DO NOT TRY TO USE THE OPTICALLY-ISOLATED SWITCHES WITH ALTERNATING CURRENT ELECTRICITY. IN PARTICULAR, DO NOT TRY TO USE THE SWITCHES WITH NORMAL (120 VOLT AC) HOUSEHOLD ELECTRICITY. 120 VOLT AC WILL DESTROY THE OPTICALLY-ISOLATED SWITCHES AND MAY DAMAGE YOU OR THE COMPUTER.

TROUBLESHOOTING

The following steps should help you locate a problem in the switch circuit. Proceed through these checks in order until unexpected results are encountered. This should help you "zero-in" on the source of the problem.

1. Is the LED in the optoisolator turning on? Attach a voltmeter's negative lead to the Apple computer's GND lead (pin #7 of the 7404). Run the FLASHER program while touching the positive lead of the voltmeter to pin #14 of the 7404. The voltage here should be a steady 5 volts. Now touch the positive lead of the voltmeter to pin #2 on one of the optoisolators. The voltage at that point should periodically change from near 0 to near 5 volts. If the voltage does not change, there must be something wrong with the 7404 circuit.

2. Is the phototransistor inside the optoisolator turning "on"? Attach a voltmeter's negative lead to the "-" terminal of the switch. Connect an external circuit to the switch and turn on the external power supply. Run the FLASHER program. Touch the positive lead of the meter to pin #5 of the optoisolator. The voltage at this point should periodically change from near zero to near the power-supply voltage. If it does not, the phototransistor is not wired properly. Note: This test will not work unless an external circuit (with the power supply on) is connected.

3. Are the transistors turning on? Attach a voltmeter's negative lead to the "–" terminal of the switch. Connect an external circuit to the switch and turn on the external power supply. Run the FLASHER program. Touch the positive lead of the meter to the collector (C) lead of the TIP-3055. The voltage at this point should periodically change from near zero to about 0.7 volts. If not, double-check the wiring of the transistors. Note: This test will not work unless an external circuit (with the power

supply on) is connected. If this test fails, and the wiring is correct, try substituting new transistors. Transistors can be damaged fairly easily by overheating during soldering.

Project Extensions

Program Improvements

◊ Modify the FLASHER program so it counts in binary. Designate the LED that is attached to switch 0 as the units place. The other LEDs represent 2, 4 and 8. Rework the FLASHER program so the LEDs count out from 0 to 15 in binary as shown below. There are a lot of different ways to write this program. Try to use a method that involves using a pattern. Do not just use 16 statements, each containing four poke commands.

● LED ON
○ LED OFF

	SWITCH					SWITCH			
	3	2	1	0		3	2	1	0
0	○	○	○	○	8	○	○	○	●
1	○	○	○	●	9	●	○	○	●
2	○	○	●	○	10	○	●	○	●
3	○	○	●	●	11	●	●	○	●
4	○	●	○	○	12	○	○	●	●
5	○	●	○	●	13	●	○	●	●
6	○	●	●	○	14	○	●	●	●
7	○	●	●	●	15	●	●	●	●

◊ Write your own programs to control the Optically-Isolated Switches. Use the program to control DC lamps, motors, radios, fans or motors. It can be designed to respond to keyboard instructions or signals from a photogate (Project 1) or a microphone/amplifier (Project 3).

◊ If you have software or hardware with your computer system that allows voice recognition, teach the program to turn on and off electrical circuits at your voice command. Systems like this are sometimes used by severely handicapped people.

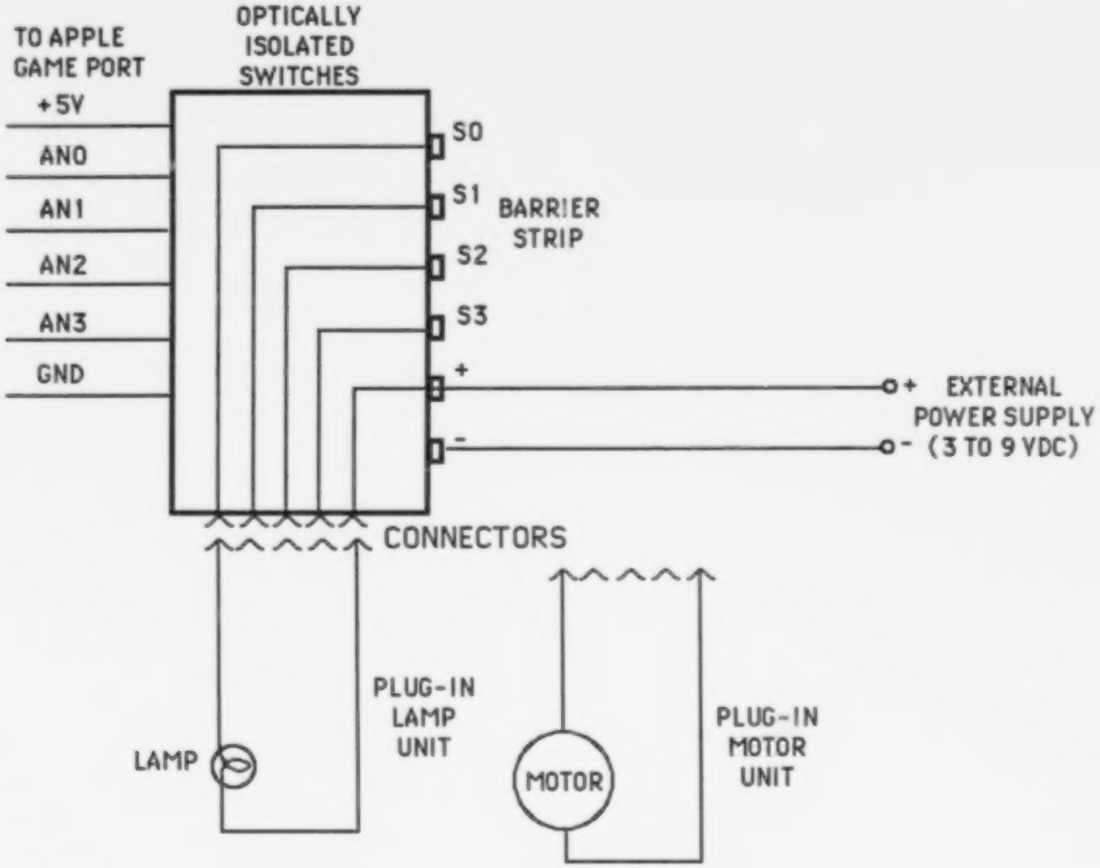
Circuit Modifications

Three of the projects that follow in this book use the Optically-Isolated Switches as a starting point: *Computer-Controlled Car*, *Stepper Motor* and *A Better Mousetrap*. Here are some other circuit modifications you might want to try:

◊ Mount the Optically-Isolated Switches in a box. Choose a box that has room for the circuit board inside and is large enough for mounting the barrier strip on the outside.

◊ If you ever plan to use the switches with devices that draw large currents (mainly devices that generate a lot of heat), you may want to add extra *heat sinks* to the TIP-3055 transistors. Heat sinks are pieces of metal that are bolted onto the metal part of the transistor to help get rid of the heat produced in the transistor. If you find that the TIP-3055 transistors get hot, add heat sinks. Heat sinks for this type of transistor are called TO-220 heat sinks. They are sold by most electronic-supply houses. Two versions sold by Radio Shack are RS #276-1367 and RS #276-1363.

◊ If you plan to use the Optically-Isolated Switches for a number of different uses, you may want to use a connector that allows you to quickly plug different circuits into the switches. Some possibilities are sub-D connectors, DIN connectors or headers. You will need five leads on the connectors. A sample connection using only S0 is shown below:


TO APPLE GAME PORT
OPTICALLY ISOLATED SWITCHES
+5V
AN0
AN1
AN2
AN3
GND
S0
S1
S2
S3
+
-
BARRIER STRIP
+ EXTERNAL POWER SUPPLY
- (3 TO 9 VDC)
CONNECTORS
PLUG-IN LAMP UNIT
LAMP
MOTOR
PLUG-IN MOTOR UNIT

Use the same connector on all of the other circuits you build that use the Optically-Isolated Switches. Notice that with this connection system, both leads of the power supply are connected to the barrier strip and no separate lead from the device to the power supply is needed.

◊ Connect the Optically-Isolated Switches to a heat source and combine it with the IC Temperature Probe (Project 6) to make a controlled temperature environment. For example, use a DC lamp as a heat source inside a box. Place the IC Temperature Probe inside the box. Write a BASIC program that turns on the lamp whenever the temperature in the box drops below a certain level. Have the program turn off the lamp when the temperature exceeds a limit. There are a number of practical uses for this kind of device, including egg incubators and controlling the temperature for paint or fruit drying. Two warnings: (1) Some heat sources draw a lot of current. Make sure the switch circuit does not get overheated by running too much current through it; and (2) Be careful not to start a fire with the heat source.

◊ Set up the Optically-Isolated Switches as an alarm system. Connect the Optically-Isolated Switches to a siren or a bell. Combine this hardware with a photogate or another type of switch to make an alarm system. Write a program that waits for a signal at the detector and then turns on the warning devices. Most alarm-system warning devices work on 12VDC, so they will work with the Optically-Isolated Switches.[11]

---

[11] The *Apple Connection* by James W. Coffron has a chapter on using an Apple computer as the heart of a home-security system.

◊ Combine the Optically-Isolated Switches and a photogate to make a "monkey gun." A popular physics demonstration involves a "blow gun" that fires a projectile at an object suspended from an electromagnet. The target is released by the electromagnet at the instant the gun is fired. One Optically-Isolated Switch can be used to control the electromagnet. A photogate similar to project #1 can be used to detect the moment of the projectile launch.[12]

SUPPLEMENTAL INFORMATION

Technical Information About the Optically-Isolated Switches

This switch circuit is based on a circuit in *The Discrete Transistor Circuit Sourcemaster* by Kendall Webster Sessions, John Wiley and Sons, 1978. The same circuit is mentioned in a *Byte* article by Steve Ciarcia, December 1984. The Ciarcia article is a good reference for information on controlling all sorts of devices (including 120-volt AC appliances) with a computer.

The circuit can be used to switch DC currents of up to 10 amperes. If the circuit is going to be used for extensive periods with large currents, heat sinks should be bolted to the TIP-3055 transistor to help prevent overheating (see the "Project Extensions" for details). The voltage across the switch should not exceed 25 volts DC.

When the computer is first turned on (or rebooted), some of the Optically-Isolated Switches will be turned on. Keep this in mind as you use them.

When the switch is operating and in the "on" state, there will always be at least a 0.7 voltage drop across the switch. This potential difference increases slightly as the current increases. Keep this voltage drop in mind when you plan on power supplies. For example, if you are controlling motors that must have 5.0 volts to operate properly, your power supply will need to supply nearly 6 volts to allow for the voltage drop across the switch.

Program Design Notes

All of the programs on the *Project Programs* disk that use the Optically-Isolated Switches use the following constants for the POKE statements. These statements turn on and off the switches as follows:

N0 = 49240 (used to turn S0 off)
C0 = 49241 (used to turn S0 on)
N1 = 49242 (used to turn S1 off)
C1 = 49243 (used to turn S1 on)
N2 = 49244 (used to turn S2 off)
C2 = 49245 (used to turn S2 on)
N3 = 49246 (used to turn S3 off)
C3 = 49247 (used to turn S3 on)

For example, a typical statement to turn on the S0 switch could be written:

```
100 POKE C0,0 : REM CLOSE SWITCH 0
```

The program SWITCH uses one other technique that may require some explanation. In this program, four variables (S0, S1, S2 and S3) are used to indicate and control the current status of each of the switches. If the value is zero, then the switch is turned off. If the value is one, the switch is on. To change the status of a particular switch, the program uses a statement like this:

---

[12] See "Long-Range Apparatus for Monkey and Hunter Demonstration" by Akio Saitoh, *The Physics Teacher*, Volume 19, #8, p. 563.

```
200 S0 = NOT S0
```

The "NOT" is an Applesoft BASIC command that assigns a 0 to S0 if it was 1, or assigns a 1 to S0 if it was 0.[13]

---

[13] This is an example of Boolean function. Others include "AND", "OR", "NOR", etc.

PROJECT 12:
COMPUTER-CONTROLLED CAR

In this project you will build a circuit to allow the computer to control two DC motors. Each motor can be controlled separately. These two motors will power the rear wheels of a small car. Using the appropriate software, the car can be made to travel forward, backward, left or right. The Optically-Isolated Switches from Project 11 are required. This project is a good starting point for experimenting with robotics. The project extensions include building collision detectors and light sensors into the car, and measuring the amount of movement of the car.

PROJECT REQUIREMENTS

Project Prerequisite

The Optically-Isolated Switches from Project 11 are needed for this project.

Background Knowledge

- General: Putting together the car requires some construction skills.
- Electronics: The circuit is not very complex, but experience with soldering is recommended.
- Programming: None required for the core project; some of the project extensions require experience with BASIC programming.

Parts

☐ (2) 5-volt DC motors with a gear speed-reduction system that can be used to power the rear wheels of a toy car (Several electronic-parts suppliers sell a "Dual Drive Gear Motor Unit," which includes two motors and the necessary gears. The unit is a surplus part from a toy manufacturing project. It operates on 5 or 6 volts and is ideal for this project. It includes a gear system that provides slower rotation and more power to the rear wheels. This unit sells for about $5.00 and was available in the summer of 1986 from All Electronics, 905 S. Vermont Ave., P.O. Box 20406, Los Angles, CA 90006, (213) 380-8000. It is their part

#DCM-R. The same unit, called a "robot motor" [#J31,827] is available from Edmund Scientific, 101 E. Gloucester Pike, Barrington, NJ 08007 (800)-222-0224. The project can be done with any two DC motors and without a gear system, but in general this will yield too much speed and not enough torque.)

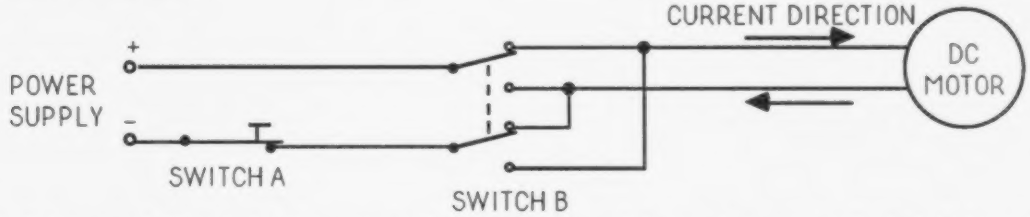
- ☐ toy car body with pivoting front wheel (We recommend going through your little brother's or sister's toy box when he or she is not looking. Another option is to visit a thrift store to purchase an inexpensive, used toy car, truck or tractor.)
- ☐(2) 5-volt double-pole, double-throw relays with contacts rated at a current of 300 mA (These are available from many electronic-parts suppliers, including Radio Shack [#275-244], Mouser Electronics [#ME433-MV2P5VDC] and Digi-Key [#Z313-ND].
- ☐ circuit board (Use any circuit board with room for two 16-pin DIP ICs. A Radio Shack Dual IC Circuit Board [RS #276-159] works well.)
- ☐ 4-conductor wire to run from the controller to the car (Four leads from a ribbon cable may be used; another possibility is telephone cable. The wire should be light and flexible and long enough to let the car move around on the floor.)
- ☐ miscellaneous connection wires for the circuit

Tools and Supplies

Epoxy glue, solder, soldering iron, multimeter, pliers, screwdriver, plastic electrical tape, wire cutters and wire strippers. You will also need a 6-volt DC power supply. If available, use a "plug-in" power supply that will supply enough current at 6 volts. The Dual Drive Gear Motors Unit normally draws about 120 mA, and more when under a load. The same power supply needs to supply current for the two relays. The total current can be as high as 500 mA. Four "C" or "D" cells connected in series can be used.[1]

BACKGROUND INFORMATION

One nice thing about a direct-current (DC) motor is that it can run in either a clockwise or a counterclockwise direction. To reverse the direction of rotation, you change the direction of current through the coils of the motor. A simple circuit such as the one below can be used to operate the motor in either direction.

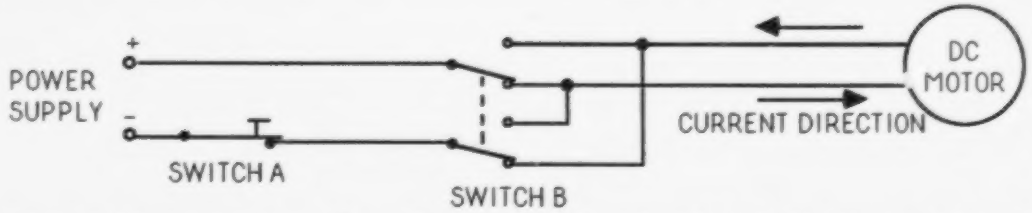


Switch A is an on/off switch. It controls the flow of electricity to the motor. Switch B is a *double-pole, double-throw* switch. It is really two switches connected together (hence "double pole") and there are two different points that can be connected to the central terminal of the switch (hence "double throw"). Assume that switch A is closed and switch B is in the position shown in the diagram above. Trace the path of positive current through the circuit. Notice that current flows downward through the motor as it is drawn in the diagram.

Now consider what happens when switch B is shifted to the other position.

---

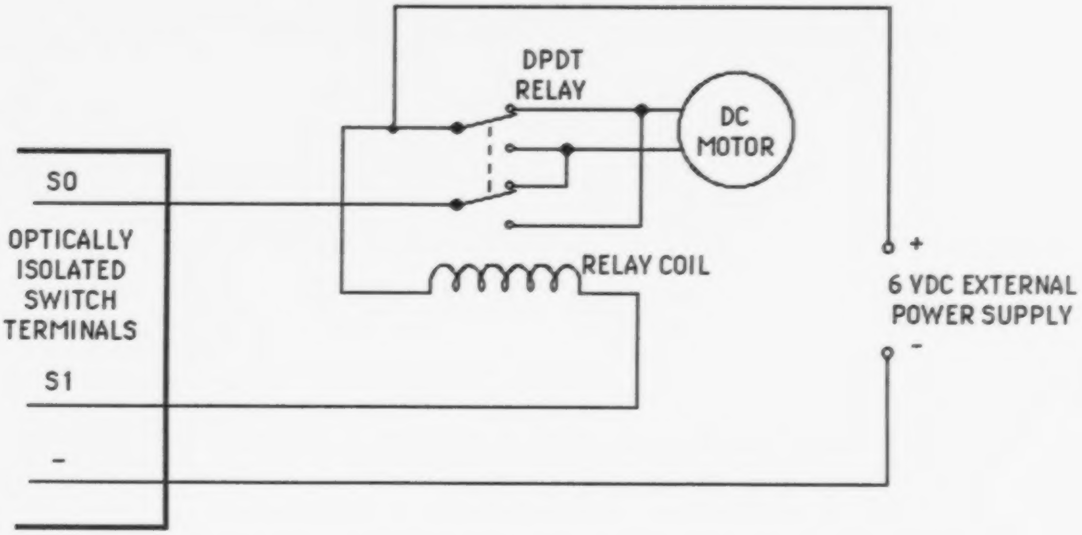
[1] You may be wondering why we are using a 6-volt power supply to operate 5-volt motors. There is a 0.7 volt (or slightly larger) potential drop in the Optically-Isolated Switch circuit. That means a 6-volt power supply will deliver just about the right potential to the motors.



Notice that positive current now flows in a upward direction though the motor as drawn in the diagram.

A computer can be used to operate the switches needed for DC motor control. The Optically-Isolated Switches (from Project 11) are ideal for this job. The switches used for this project must be able to handle the fairly large currents that the motors and the coil require. They must also be able to handle inductive loads.[2] Both of these requirements are met by the Optically-Isolated Switches.

One of the Optically-Isolated Switches can be used as the on/off switch (switch A). Another one of the switches can be used to control a *relay* to accomplish the job of switch B. A relay is an electrically-controlled switch in which the connecting arms are moved by the pull of an electromagnet. Relays are available in many different forms, including double-pole, double-throw (DPDT) configurations. The diagram below shows the circuit we will use to control the direction of the current in one motor.


DPDT RELAY
DC MOTOR
S0
OPTICALLY ISOLATED SWITCH TERMINALS
S1
-
RELAY COIL
+
6 VDC EXTERNAL POWER SUPPLY
-

SCHEMATIC DIAGRAM OF THE MOTOR-CONTROL CIRCUIT

Switch S0 is the on/off switch. Switch S1 controls the direction of the current. Whenever S1 is closed, current flows in the coil of the relay. This causes the coil to become an electromagnet and moves the connecting arms of the switch closer to the coil. When this current stops, a spring forces the poles of the switch to return to their original position (the position shown in the diagram).

Notice that an external power supply is required for two purposes in this project. It supplies the power for the motors and it also powers the electromagnet in the relay. A single 6-volt DC power supply is recommended.[3] You can use either four 1.5-volt batteries or a DC power supply that can supply about 500 milliamperes of current at 6 volts DC.[4]

---

[2] An inductive load is any device using electromagnetic fields, such as a motor or a relay. A large reverse current can be produced when a magnetic field of a coil or motor collapses. Diodes were placed in the circuit of the Optically-Isolated Switches to protect the components from this reverse current.

In this project, we will build two circuits like the one described above, and use them to drive the rear wheels on a toy car, truck or tractor. Optically-Isolated Switches S0 and S1 are used to control the left wheel and switches S2 and S3 are used to control the right wheel Any two DC motors could be used to power the wheels of the car, but we recommend you use motors with a speed-reduction gear system. Without a gearing system on the motors, the car will probably run too fast and have too little torque. Several electronic-parts suppliers sell a surplus toy part that is ideal for this project. This unit contains two 5-volt DC motors, each of which is connected to a gear system and then to an axle. With this unit, all you need to do is add wheels to the rear axles and a front end to complete the car. Information on where to get this unit is given in the parts list for this project.

The MOTOR program on the *Project Programs* disk controls the four switch output lines to allow the car to perform the following operations:

key pressed	operation	left motor	right motor
F	FORWARD	FORWARD	FORWARD
R	RIGHT	FORWARD	BACKWARD
L	LEFT	BACKWARD	FORWARD
B	BACKWARD	BACKWARD	BACKWARD
S	STOP	OFF	OFF

Using this program, the car can be controlled as it moves around the room. The details of how the MOTOR program operates are covered in the "Supplemental Information" section.

STEP-BY-STEP INSTRUCTIONS

☐ 1. **Attach wheels to the axles of the car.** The computer-controlled car should have two independent, motor-driven rear wheels. We recommend using the "dual-drive gear motors" that are listed on the parts list. If you use this unit, the motors and the gearing are taken care of and all you need to do is find wheels to glue onto the axles. Explore local toy boxes or thrift stores to find an old toy with wheels that can be used for this purpose. As you look for a toy to sacrifice, remember that you will also need a "front end" for the car you are building (see step #2 below). Carefully glue the wheels to the axles. Use a strong glue, such as epoxy, and allow it to dry thoroughly.

☐ 2. **Add a pivoting front wheel to your car.** Use your ingenuity here. You may be able to find a used toy car that you can cut in half and glue to the drive unit. Otherwise build a front end for the car. The front wheel or wheels on the car should pivot to allow the car to easily turn left or right.

☐ 3. **Connect lead wires to the two motors.** Long lead wires are recommended, since you will probably want to operate the car at some distance from the computer. Use light-weight, flexible, four-conductor wire. Solder two of the wires to each motor. Insulate the solder joints carefully with electrical tape. Provide some sort of *strain relief* for the wire; that is, tie, tape or glue the wire to the car body itself. This way, if there is a pull on the wire, the individual soldered wires to the motor will not be pulled loose.

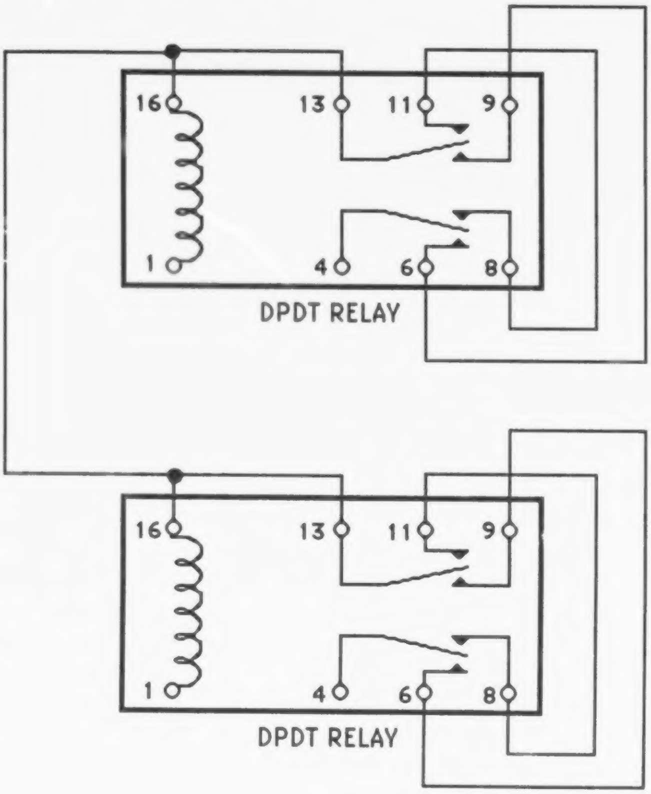
---

[3] You could use one power supply for the motors and a second power supply of a different voltage for the relays. You might do this, for example, if the only relays you can obtain have coils that operate on 12 volts and you want 5 volts for the motors.

[4] The reason we recommend a 6-volt power supply to operate 5-volt devices is due to the nearly one volt potential drop in the Optically-Isolated Switch circuit.

☐ 4. **Place the two DPDT relays onto the circuit board.** These two relays are the only components (other than wires) that are to be placed on the circuit board. Place them side by side, oriented in the same direction, some distance apart. If you are using a Radio Shack Dual IC Circuit Board, place one relay on each half of the board. Solder the pins to the plated side of the circuit board.[5]

☐ 5. **Add jumper wires to the relays.** Carefully study the pin placement on the relays. If you are using one of the relays recommended on the parts list, there should be 8 pins on each relay. The pattern of pins is as shown in the diagram below.[6] This diagram is a top view of the relay. On each relay, add three short wires connecting pin #6 to #9, #8 to #11, and #13 to #16. Then add a wire connecting the coils (pin #16) of the two relays.



☐ 6. **Connect the relays to the motors, the power supply and the Optically-Isolated Switches.** The diagram below shows the complete motor-control circuit. Two wires from each motor should be attached to each relay as shown. Make sure that the relay connected to the left wheel is also connected to switches S0 and S1 on the Optically-Isolated Switches. You will have to take a chance as to which of the two wires from each motor goes where on the relay (pin #9 versus #11). There is a 50% chance that you will have to switch them later. The only way to find out which way they go is to test the assembled unit. If you guess wrong, the wheel will go backward when it should go forward, and you will have to switch the leads.

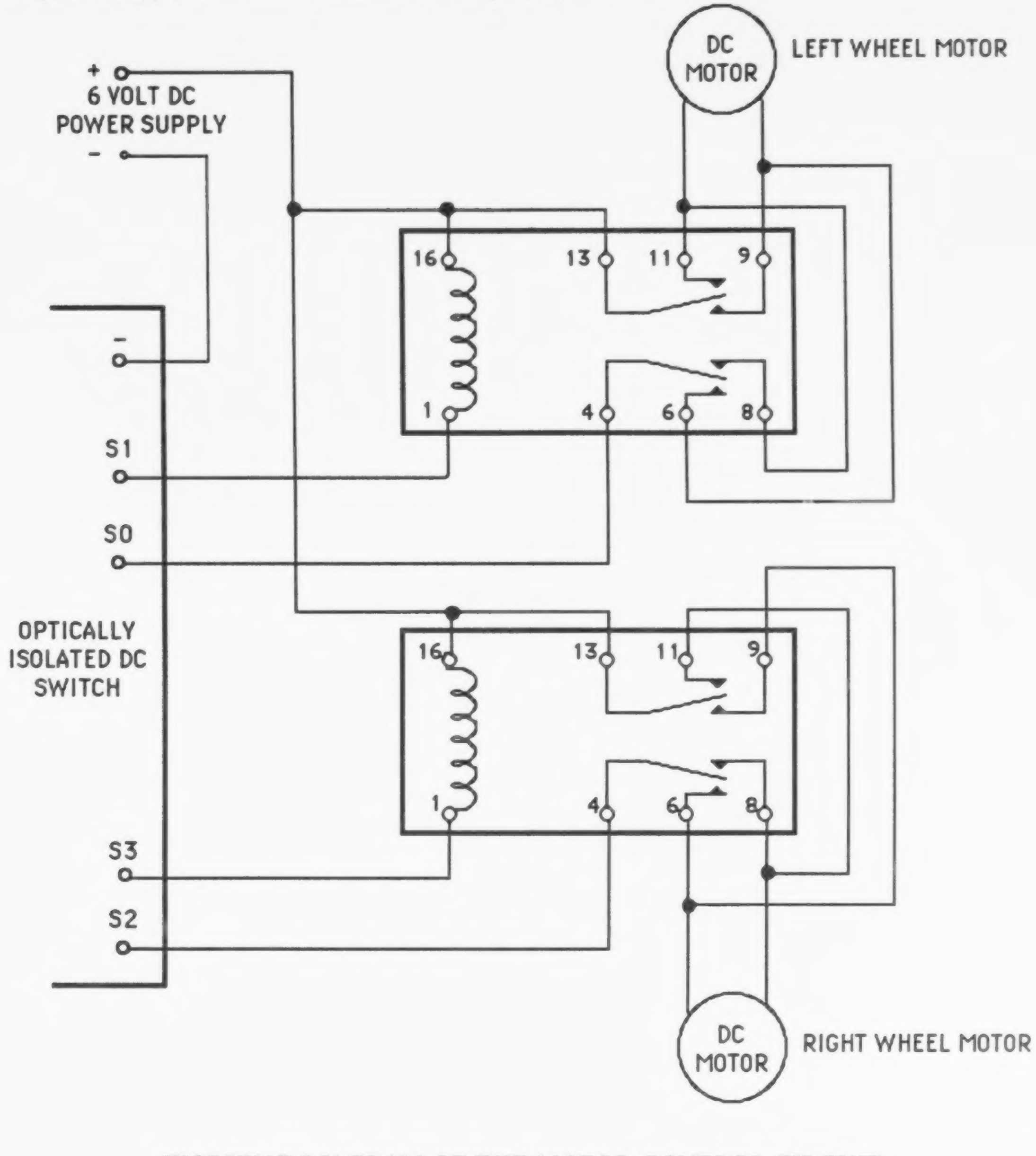
---

[5] You can use 16-pin IC sockets with these relays. Solder the sockets in place, and then plug in the relays. Using sockets will allow you to remove the relays easily if you ever want to do so.

[6] The functional parts of the inside of the relay are shown in this diagram also. If you are using a different DPDT relay, you should be able to determine which pins to use on your relay by comparing it to this diagram.

Add four wires to lead from the relays to the terminals of the Optically-Isolated Switches. Take some care to make sure the correct relay pin is connected to each switch terminal, or the motor-control commands will be confused.

The power supply for this project should be able to supply about 500 mA at 6 volts DC. With the power supply turned off (or the batteries removed from their holder), connect the two power-supply leads. One goes from the "+" terminal to pin #16 of both relays. The other goes from the "-" terminal on the power supply to the "-" terminal on the Optically-Isolated Switches.



DC MOTOR
LEFT WHEEL MOTOR
+
6 VOLT DC POWER SUPPLY
-
16
13
11
9
1
4
6
8
-
S1
S0
OPTICALLY ISOLATED DC SWITCH
16
13
11
9
1
4
6
8
S3
S2
DC MOTOR
RIGHT WHEEL MOTOR

WORKING DIAGRAM OF THE MOTOR CONTROL CIRCUIT

☐ 7. **Connect the Optically-Isolated Switches to the computer.** With the computer off, connect the Optically-Isolated Switches (which are connected to the control circuit) to the game port. Run the program MOTOR on the *Project Programs* disk.

☐ 8. **Test the project.** Place the car where it can move freely, turn on the power supply, and press the "F" (FORWARD) key. Note how each wheel behaves. If either (or both) wheels move backward instead of forward, you need to switch the two leads for that motor where they attach to the circuit board. If the motors do not run at all, refer to the "Troubleshooting" section below.

☐ 9. **Try out the other control keys.** Try pressing "B", "L","R" and "S." Notice how the car operates. In the L (LEFT) mode, the right wheel should turn forward, and the left wheel should turn backward. The R (RIGHT) mode is the opposite.[7] If the wheels do not turn as they should, refer to the "Troubleshooting" section.

TROUBLESHOOTING

If you have problems with this project, first check for the most likely errors:

- Make sure the Optically-Isolated Switches are connected correctly and working properly. Connect a simple test circuit to the switch unit and test it with the SWITCH program. Make sure all four switches are turning on and off as expected.
- Check for a wiring error, especially with the leads on the DPDT relays.
- If one of the motors operates when it should, but always in the opposite direction, the two leads from the control circuit to the motor are reversed. Switch the leads at the points where they connect to the relay.
- Make sure your external power supply is on and working properly.

◊ If a motor will not turn on in any mode of operation

If the Optically-Isolated Switches are working properly and the motor will not run, do some troubleshooting with a voltmeter. Any voltmeter that will read DC voltages between 0 and 5 volts can be used. Connect the negative lead to the "-" terminal on the power supply. The voltmeter tests should be done with everything set up and with the computer running the MOTOR program.

- Check to make sure the power-supply leads are connected to the relay. Touch the positive lead of the voltmeter to pin #16 of the motor (refer to the diagrams in the "Step-by-Step Instructions" for pin identification). The voltage here should be about 6 volts DC (the power-supply voltage). If it is not, either the power supply is not working, or there is a problem with the wiring. There are three wires to check: (1) the wire from the "+" terminal on the power supply to the relay (pin #16), (2) the wire from the relay (pin #1) to the appropriate "S" terminal on the Optically-Isolated Switches, and (3) the wire from the "-" terminal on the power supply to the "-" terminal on the Optically-Isolated Switches.
- Check to see if power is getting to the motor. Press the "B" (backward) key, so the motor turns on.[8] Touch the positive lead of the voltmeter to each of the two leads of the motor in question. One lead should be at about 6 VDC and the other should be at less than one volt. If that is not the case, double-check the leads to the motor and the wiring of pins #4, #6, #8 and #10 on the relay.

◊ If a motor will not switch directions

This indicates that either the relay is not switching or the wiring of the switch part of the relay is wrong. With the circuit connected, run the MOTOR program, put your ear near the relay, and press the "L" and "R" keys several times. If the coil part of the relay is wired correctly, you should hear a "click" as the relay switches. If you do not hear the sound, double-check the wiring of the relay coil (pins #1 and #16). Also, make sure you are using a relay that operates on 5 volts (some relays need 12 volts). If there is a click, the problem must be in the wires that connect to the motor.

---

[7] Since relays are mechanical switches, they take a little time to respond. This can momentarily cause unusual movement when the car switches modes of operation.

[8] The "B" mode is the best to use because it turns on all four switches. Even if you have the switch leads mixed up, this should turn on the motor.

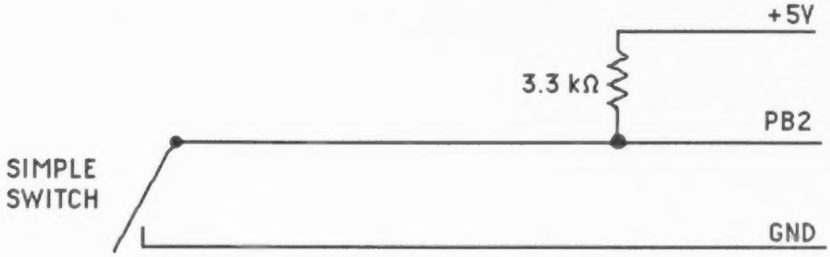
◊ If the control commands (L, R, F, B, and S) are all mixed up

Study the pattern of what direction each wheel turns when the various command keys are pressed. If the problem is that a particular wheel always goes in the wrong direction, simply reverse the two leads to that motor. If the pattern is more complicated, the most likely problem is a mix-up in the leads between the relays and the Optically-Isolated Switches.

PROJECT EXTENSIONS

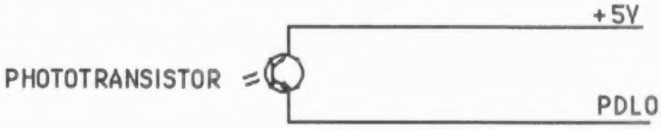
Circuit Modifications

◊ Add a collision sensor to the car. Make up a mechanical switch to mount on the front of the car. The switch can be as simple as a bent paper clip that touches another wire to complete a circuit when it pushes against something. Connect this switch to the Apple game port as shown below:[9]



After you have added this collision detector to the car, try rewriting the MOTOR program so the car takes "evasive action" whenever it bumps into something. For example, you may want it to back up a little, turn left and then proceed.

◊ Add a light sensor to the car. A phototransistor, such as the one used in the photogate (Project 1), can be used as a light sensor. Connect it as shown below:



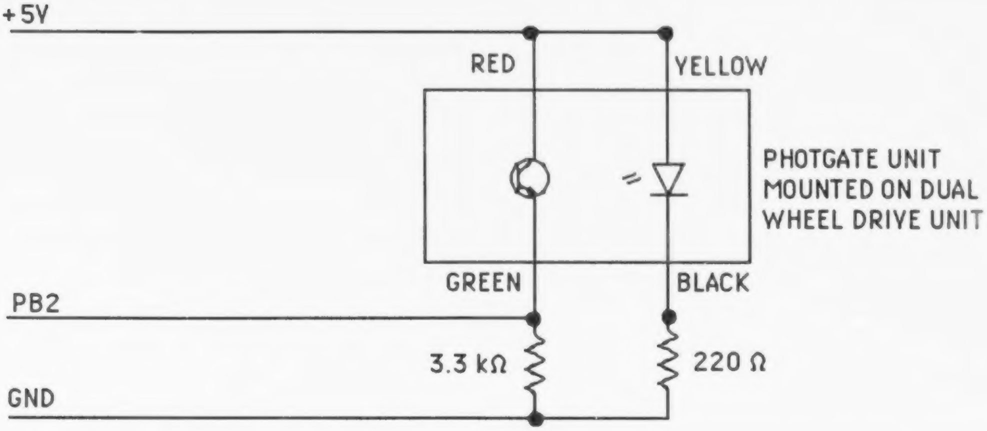
The light intensity can be determined using the BASIC PDL(0) command.

◊ Combine the two on/off switches. Two of the Optically-Isolated Switches (S0 and S2) in the core project are used as on/off switches. You could use the same switch to connect or disconnect power to both motors. This will "free up" one of the Optically-Isolated Switches for some other use. The only thing you lose is the ability to turn off one of the motors at a time.

◊ Add an odometer to the car. The Dual Drive Gear Motor Unit recommended for this project usually comes with a miniature photogate, which is designed to be used to detect the number of turns of a gear within the car. If this photogate is wired to produce a voltage variation on the PB2 line, you can count pulses from this input line and count the number of pulses and calculate the distance traveled. A circuit to use is shown below:

---

[9] For this and other project extensions, you will need to connect to the game-port cable that was used in building the Optically-Isolated Switches.



In order to count these pulses, you may need to write a machine-language program; BASIC may not be fast enough. Counting the pulses from the photogate can allow you to measure distance traveled. Counting the frequency of the pulses from the photogate (similar to Project 3) can allow you to measure the rotational speed of the motor and the car speed.

◊ Build a solid-state control circuit. Transistors can be used in place of the relays for the motor-control circuit. Some care must be taken to make sure that the transistors used can handle large enough currents for the motors. Many books on robotics include sample control circuits using transistors. One example is *Apple II/IIe Robotic Arm Projects* by John Blankenship.

◊ Use wireless remote control to the car. Radio Shack sometimes sells an inexpensive Radio-Control Transmitter and Receiver (RS #277-1012) that you may want to try to use as the connection between the Optically-Isolated Switches and the car. The method used for controlling the car with this control unit is different from the method used in the core project, so the MOTOR program will have to be rewritten.

◊ Experiment with speed-control circuits for the motors. There are many different ways of controlling the speed of a DC motor. Several of them are discussed in the article "DC Motor Control: Build a Motorized Platform" by Steve Ciarcia, *Byte*, May 1981.

Program Improvements

◊ Add new modes to the MOTOR program. The "L" and "R" modes of the MOTOR program cause the two wheels to rotate in opposite directions. A different type of turning can be done by turning off one wheel and letting the other wheel rotate forward. Change the MOTOR program to make up two new turning modes using this technique.

◊ Turn off the power to the motors during mode changes. Because the relays take a moment to switch contacts, the motors may momentarily run in the wrong direction as the mode is changed. You can eliminate this by turning off the power to the motors (switches S0 and S2) for a moment during the mode shift. This requires adding a few steps to the MOTOR program.

◊ Let the car use "feedback." If you have built collision detectors, a light sensor or an odometer onto the car, program it to use the signals from these devices to make decisions. For example, program the car to "seek light." Have the car make decisions about how to move, based on how the light intensity changes as it moves.

If you have connected the photogate that is built into the Dual Drive Gear Motors, you can write programs to measure either the distance or speed using the signal from it.

◊ Modify the program to allow the car to be controlled by a joystick. Rewrite the MOTOR program so that it checks the position of a joystick and controls the car accordingly.

◊ Write programs to move the car in specific patterns. By controlling the amount of time the car is in a turning mode, you can program car movement in turns of various angles. Use this technique to program subroutines to cause the car to execute a 90° turn left or a 90° turn right. Make up a program using these subroutines to cause the car to move in patterns that you select.

◊ Experiment with controlling the car with LOGO programming.

SUPPLEMENTAL INFORMATION

Program Design Notes

The MOTOR program on the *Project Programs* disk does the job of turning on and off the Optically-Isolated Switches in the appropriate manner to cause the car to move in certain ways. It is a BASIC program that uses a "GET" statement to detect when any one of five control keys is pressed.[10] The five keys used and the response they cause are listed in the table below:

				left wheel		right wheel	
key pressed	operation	left motor	right motor	S0	S1	S2	S3
F	FORWARD	FORWARD	FORWARD	ON	OFF	ON	OFF
R	RIGHT	FORWARD	BACKWARD	ON	OFF	ON	ON
L	LEFT	BACKWARD	FORWARD	ON	ON	ON	OFF
B	BACKWARD	BACKWARD	BACKWARD	ON	ON	ON	ON
S	STOP	OFF	OFF	OFF	OFF	OFF	OFF

BASIC poke statements are used to turn on or off the switches in the pattern shown in the table. Switches S0 and S2 are simply used to turn on the power to the motors. Both motors are on except when the stop command is given. The S1 switch controls the direction of rotation of the left wheel. If the switch is on, the wheel should rotate backwards. Switch S3 controls the direction of the right wheel in the same way.

---

[10] The program also responds to <CONTROL> <C>, which ends the program.

PROJECT 13: STEPPER MOTOR

This project involves connecting a *stepper motor* to the computer. A stepper motor is a special kind of motor, very different from a conventional motor. Instead of just turning it on or off, like a normal motor, you can precisely control the position of the rotor of a stepper motor. The Optically-Isolated Switches (Project 11) are used to connect the stepper motor to the game port. The software included on the *Project Programs* disk allows you to switch on and off the coils of the stepper motor in just the right pattern to cause the motion desired. This is a good project if you are interested in experimenting with robotics, since many robots are controlled with stepper motors.

PROJECT REQUIREMENTS

Project Prerequisite

The Optically-Isolated Switches from Project 11 are needed for this project.

Background Knowledge

- Electronics: None. If you get the right type of stepper motor and can identify the leads, this is one of the easiest projects in the book.
- Programming: None required for the core project; some of the project extensions require experience with BASIC programming. Machine language can be used for some of the project extensions.

Parts

☐ small stepper motor, 5 or 12-volt, unipolar (Make sure it is a *unipolar* or *4-phase* stepper motor. Unipolar stepper motors have four coils and at least 5 lead wires. *Bipolar* or *2–phase* stepper motors cannot be used for the core project (they can be identified because they have only 4 wires).[1] You can use either 5 or 12-volt versions. Gear-reduction systems on the stepper motor are not required, but can be used. Stepper motors can be very inexpensive ($3.00 to $5.00), if you can find them sold as surplus. If you have to order them directly from the manufacturer, they will be expensive ($40.00 or more). If you get a stepper motor and cannot identify how the leads are connected inside the motor, it may be difficult to use. The "Step-by-Step Instructions" for this project assume you are using an AirPax unipolar stepper motor. If you are using any other brand of stepper motor, make sure you get information sheets with the motor, explaining how it is to be wired.

Several possible sources for stepper motors are listed below:

- R & D Electronics, 1202H Pine Island Road, Cape Coral, FL 33909, (813) 772-1441. This is a mail-order, surplus-electronics company. They sell several small, unipolar stepper motors for $4.00 or less. Their part #17E1 is highly recommended, since it is an Airpax motor and can be wired exactly as shown in the "Step-by-Step Instructions." They have several other stepper motors. Write to them for their catalog.
- Electronic Surplus Inc., 1224 Prospect Avenue, Cleveland, OH 44115, (216) 621-1052. This is the retail store for R & D Electronics (described above).
- BNF Enterprises, P.O. Box 3357, Peabody, MA 01961-3357, (617) 531-5774. They sometimes sell surplus stepper motors.
- Airpax Corporation, Cheshire Division, Cheshire Ind. Park, Cheshire, CT 06410, (203) 272-0301. This is a stepper motor manufacturer. Any of their small unipolar stepper motors would be good for this project. If you call them, they can give you information on the sales office in your area.

☐ 5-conductor wire to extend from the stepper motor to the Optically-Isolated Switches

☐(2) single-conductor wire to extend from the power supply to the Optically-Isolated Switches and to the stepper motor

Note: A "Stepper Motor Parts Kit" containing a stepper motor and connection wire is available from Vernier Software for $8.00. Specific instructions on how to connect this stepper motor will be included. An order form is at the back of this book.

Tools and Supplies

Wire cutters, wire strippers, screwdriver. You will also need a DC power supply, 5 or 12 volts, to match the stepper motor. Stepper motors draw quite a lot of current, usually a few hundred milliamperes. If you use a commercial power supply, make sure it can safely supply the current needed for your motor. Batteries can be used as a power supply for this project. Use 4 "C" or "D" cells to supply the power for a 5-volt motor or use eight 1.5-volt cells to power a 12-volt motor.[2] Be aware that the batteries will need to be replaced or recharged fairly frequently.

---

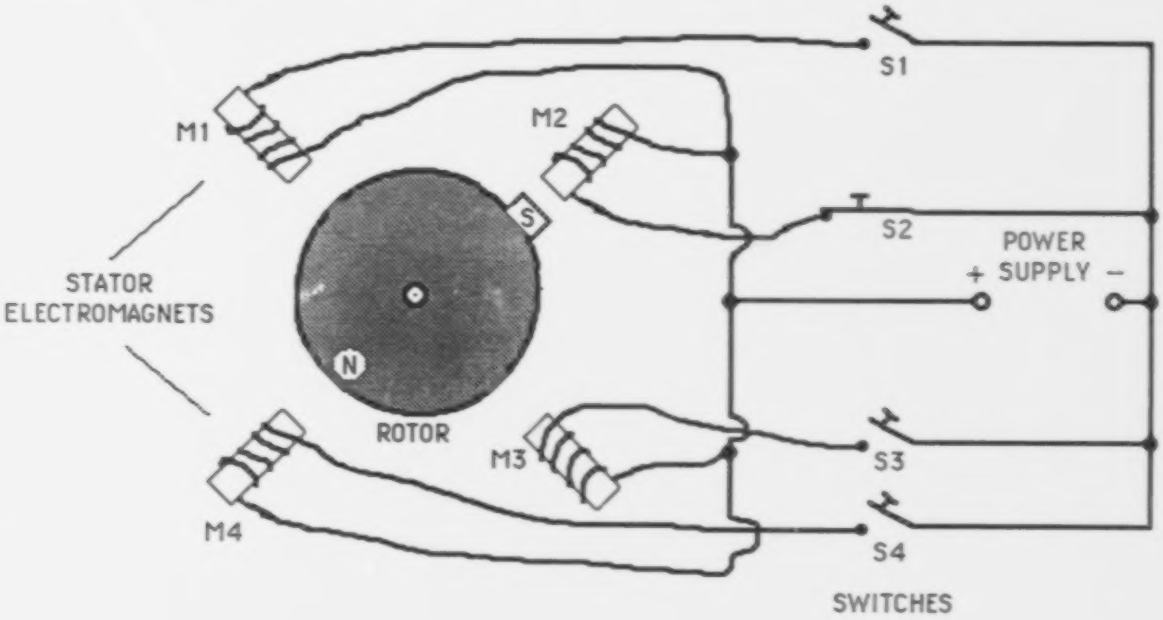
[1] One of the project extensions does explain how to connect a bipolar stepper motor to the computer. However, it is a more complicated project. The "Step-by-Step Instructions" for this project are for unipolar stepper motors only.

[2] Four "C" or "D" cells would nominally be 6 volts. However, the actual voltage supplied is never really 1.5 volts per cell, and in addition, there is a 0.7-volt drop across the Optically-Isolated Switches. As a result, four standard batteries work out about right. Eight "1.5-volt" batteries will be slightly less than the rated voltage for a 12-volt stepper motor.

BACKGROUND INFORMATION

A Simplified Explanation of How a Stepper Motor Works

Stepper motors are designed to produce rotational motion that can be precisely controlled. They are very different from conventional motors, both in the way they are built and in the way they operate. The diagram below is intended to give you an idea of how a stepper motor works.



As with a conventional motor, two key parts are the *stator* and the *rotor*. The stator is the fixed (nonrotating) part of the motor. It consists of coils of wire that become electromagnets when current passes through them. The rotor is a permanent magnet that rotates about a central pivot point. In the simple stepper motor diagrammed above, the rotor has only a single pair of magnetic poles. The stator of the stepper motor has four independent electromagnets, which can be turned on by the switches S0, S1, S2 and S3. The wiring of these electromagnets (stators) is such that when current flows through them, the end of the electromagnet nearest the rotor becomes a north pole. As shown in the diagram, S2 is closed, and the south pole of the rotor is lined up with the north end of the M2 electromagnet. The rotor will hold steady in this position because of the attraction between two unlike poles of a magnet.

If the switches are changed in the proper pattern, the rotor can be moved. For example, to rotate the rotor clockwise in the diagram, switch S2 should be opened and switch S3 closed. The south pole of the rotor is no longer attracted to M2; it is now attracted to M3. The motor moves a single "step" when these two switches are changed. The angle through which it moves is called the *step angle*.

The rotor of the hypothetical stepper motor above could be made to rotate in a clockwise fashion by using the following switch patterns, one after the next:

	S0	S1	S2	S3
1	OFF	ON	OFF	OFF
2	OFF	OFF	ON	OFF
3	OFF	OFF	OFF	ON
4	ON	OFF	OFF	OFF
		etc.		

FOR CLOCKWISE MOTION

If this pattern is repeated, the motor will keep running clockwise. The opposite pattern causes the rotor to move counterclockwise:

	S0	S1	S2	S3
1	OFF	ON	OFF	OFF
2	ON	OFF	OFF	OFF
3	OFF	OFF	OFF	ON
4	OFF	OFF	ON	OFF
		etc.		

FOR COUNTERCLOCKWISE MOTION

This explanation of how a stepper motor operates is an oversimplification; real stepper motors are more complex than this. Both the rotor and the stator have more pairs of magnetic poles and the step angle is much smaller. However, the essential principles of stepper motors are listed below:

1. A magnetized rotor is moved a specific step angle each time the switches are changed.
2. A logical pattern of switch settings must be followed to cause the rotor to move. There is a different pattern for clockwise motion and for counterclockwise motion.
3. The rotor will hold its position if the switches are left unchanged and if the current to the electromagnets is left on.
4. By counting the number of steps that the motor moves clockwise and counterclockwise, and knowing the step angle, you can keep track of the orientation of the rotor.
5. Continuous rotation is possible if the correct pattern of switch settings is repeated. Stepper motors do not rotate as fast as conventional motors. Typically, the maximum speed is a few hundred steps per second.

If you are interested in learning more about the inner workings of a real stepper motor, refer to the "Supplemental Information" section below.

Stepper motors are sold in many different sizes, voltages and torque ratings, and with several different wiring systems. Some of them have gear-reduction systems attached to the rotor so that the effective step angle is very small. This reduces the maximum speed of rotation, but it increases the power and allows for very exact positioning. Stepper motors are popular for robotics, since they allow you to keep track of the exact position of the rotor and devices connected to it.[3] Another common use is in computer systems. They are used to position the read/write head on the disk drive at the correct distance from the center of the disk. The sound you hear as the disk drive operates is probably a stepper motor. They are also used in many printers.

Stepper-Motor Controllers

To use a stepper motor, you must have a method of operating the switches efficiently. The standard way of doing this is to use an integrated-circuit called a *stepper-motor controller*. The controller contains the switches and other electronic circuits that make controlling the motor easier. Typically, one input line on the controller is used to determine the direction of rotation of the rotor. Another input line is used as a *trigger*. The IC "knows" how to operate the four switches to cause the proper motion. The rotor on the stepper motor advances one step angle each time a pulse appears on the trigger line.

For this project, we will use the Optically-Isolated Switches and the Apple computer as the stepper-motor controller. The Optically-Isolated Switches will be closed and opened in the proper order to cause the motion desired. The program STEPPER on the *Project Programs* disk does most of the work. The program allows you to enter the number of steps that you want to rotate, and when you press <RETURN> the rotor will move. You can use positive numbers for clockwise rotation and negative numbers for counterclockwise rotation.

---

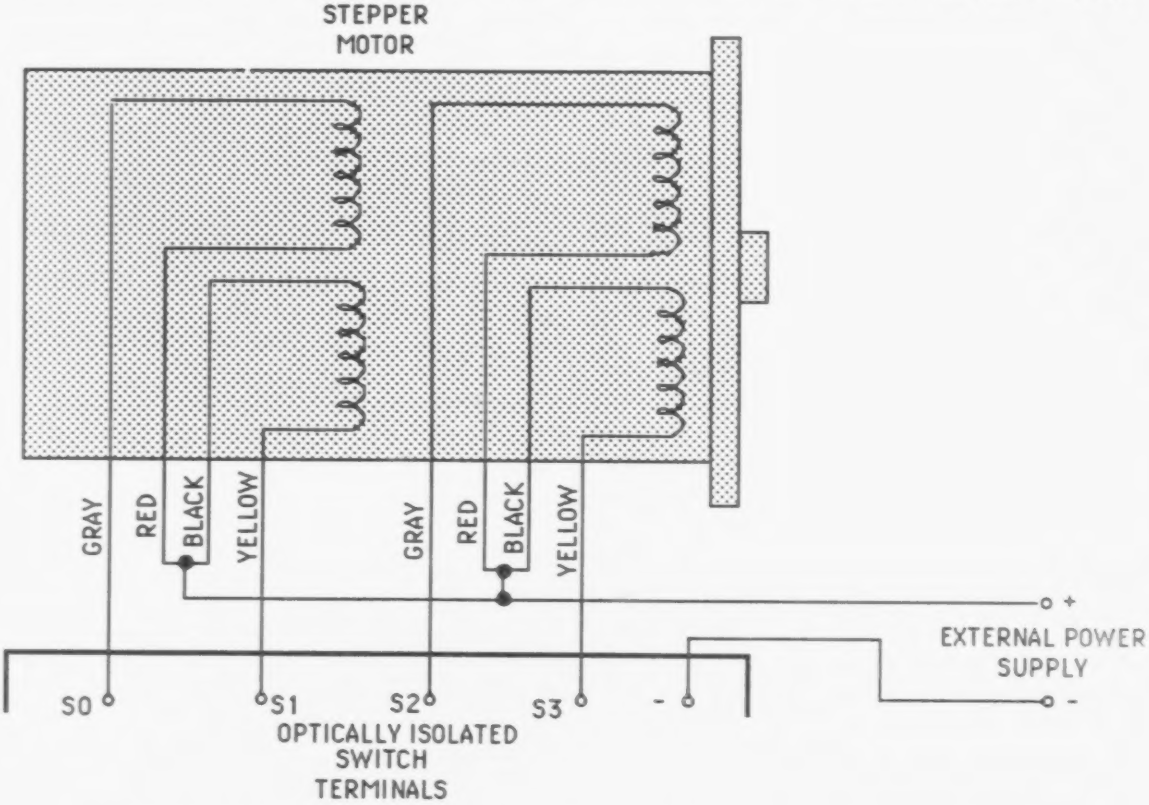
[3] There are many books available on robotics, for example, *Apple II/IIe Robotic Arm Projects* by John Blankenship.

The STEPPER program actually allows for three different drive methods for stepper-motor control: Normal, Half-Step and Wave. The sequence of switch settings that is used is slightly different in the different drive methods. The three drive methods are discussed in the "Supplemental Information" section of this project.

Many project extensions are possible. You can add new features to the STEPPER program, try out different controllers with your stepper motor, or turn your stepper motor into a robot.

STEP-BY-STEP INSTRUCTIONS

☐ 1. **Connect the leads of the stepper motor to the Optically-Isolated Switches.** The stepper motor you are using should have at least 5 (and probably 6 or 8) lead wires.[4] The diagram below will help you identify the leads if you are using an Airpax unipolar stepper motor. Note the color code used to identify the wires. The wires on an Airpax stepper motor leave the motor case in two separate bunches. The wires in the bunch farthest from the shaft of the motor should be connected to switches S0 and S1. The wires in the bunch closest to the shaft connect to S2 and S3. All of the red and black wires from both bunches will connect to the "+" lead on the power supply.



WORKING DIAGRAM OF AN AIRPAX UNIPOLAR STEPPER-MOTOR CIRCUIT

If you are using a stepper motor with a different pattern of lead wires, study the information sheets included with the motor to determine how the various wires are connected inside your motor. There should be four wires that go to the four coils inside the motor. These four wires connect to the four

---

[4] If it has only four wires, it is a bipolar stepper motor and cannot be wired as shown below. Refer to the "Project Extensions" for a method of wiring this kind of stepper motor.

Optically-Isolated Switches. Hopefully your information sheets will help you determine which wire goes to which switch. There should also be additional wires that go to the other ends of the stepper-motor coils.[5] These wires will connect to the "+" lead of the power supply. If you have no wiring information, use an ohmmeter to determine which leads connect across the coils and then try various combinations until the motor runs.

Note: Stepper motors contain permanent magnets. Be careful about placing the stepper motor near objects that can be damaged by magnetic fields (for example, computer disks).

☐ 2. **Connect the power supply.** A DC power supply should be used that supplies the correct voltage for the stepper motor you are using (usually 5 volts or 12 volts). The power supply must be able to supply a fairly large current (typically at least one hundred mA). Batteries can be used. Connect all of the remaining leads to the "+" terminal of the power supply. If you are using an Airpax stepper motor, these wires will be red and black.

Add a wire from the negative terminal of the power supply to the "-" terminal of the Optically-Isolated Switches.

☐ 3. **Connect the Optically-Isolated Switches to the computer.** Turn off the computer and connect the Optically-Isolated Switches to the game port.

☐ 4. **Run the STEPPER program.** Place the *Project Programs* disk in the disk drive and turn on the computer. Run the program STEPPER. Turn on the power supply. When asked to select drive mode, press "N" (for "Normal" drive method) and then <RETURN>.

☐ 5. **Try out the stepper motor.** The STEPPER program will prompt you to enter the number of steps that you want to rotate the rotor. Use positive numbers for clockwise rotation or negative numbers for counterclockwise rotation. Enter a number and try out the stepper motor. Observe the motion of the rotor carefully and make sure it moves as it should. You may want to attach a "flag" (a piece of tape) to the rotor shaft so that you can more easily observe the rotation. The data sheet for your motor (or even the case of the motor itself) should indicate the "step angle" for the motor. Calculate how many steps it should take for the rotor to go one complete rotation (360°) and see if you can make the rotor do it.

☐ 6. **Try out the other drive methods.** Quit the STEPPER program by pressing <Q>. Run the program again. This time choose the "HALF-STEP" (H) drive method. With this selection, the rotor should move in half normal size steps.

Stop the STEPPER motor again and rerun it. This time choose the "WAVE" (W) drive method. This drive method should produce steps the same size as the Normal drive method, even though a different pattern of switches is used.

TROUBLESHOOTING

If you have problems with this project first check for the likely errors:

◊ Make sure the Optically-Isolated Switches are connected correctly and working properly. Connect a simple test circuit to the switch unit and test it with the SWITCH program. Make sure all four switches are switching on and off as expected.

---

[5] There are usually four such wires. In some cases, these wires may be connected together internally, so there are fewer wires leaving the motor.

◊ Make sure the power supply is connected correctly and is on. Use a voltmeter, if necessary, to make sure the proper voltage (5 or 12 volts, depending on the stepper motor) is being supplied.

◊ Check for a wiring error, especially involving the leads from the coils of the stepper motor. If you are not sure of the wiring for your particular stepper motor, try switching the coil leads to see if that solves the problem. If the motor makes minor movements but never really gets going, it is likely that either some of the coils are not receiving power, or the order of the coil connections is wrong.

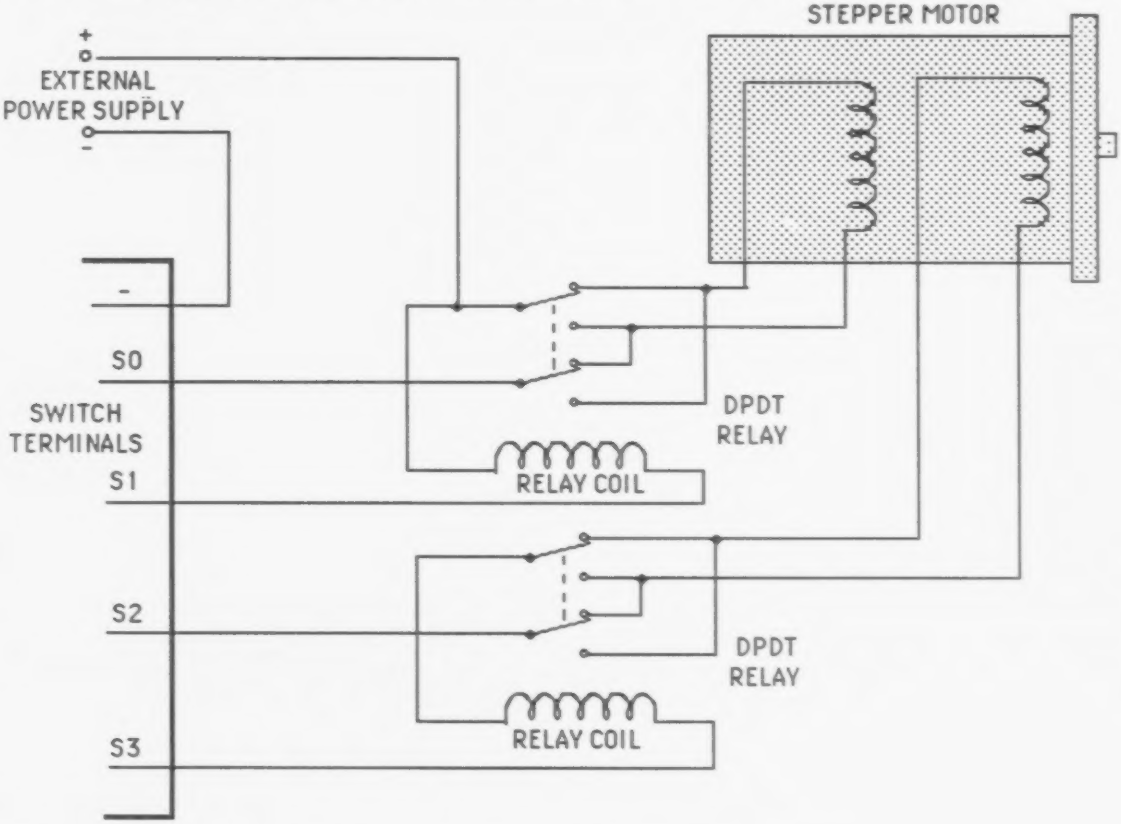
PROJECT EXTENSIONS

If you are interested in doing more experimenting with stepper motors, you should write to Airpax Corporation, Cheshire Division, Cheshire Industrial Park, Cheshire, CT 06410, (203) 272-0301 and request their *Stepper Motor Handbook*. This is their catalog of small stepper motors, but it also includes several pages of information on stepper motors, how to select them, and how they work.

Circuit Modifications

◊ Try using a bipolar (4-phase) stepper motor

Bipolar stepper motors have only two coils instead of four (as in a unipolar stepper motor). Current can be forced to flow in either direction through these coils to produce the magnetic fields needed to cause the rotation. One way to accomplish this is to use two double-pole, double-throw relays. The circuit required is very similar to the circuit used in Project 12, *Computer-Controlled Car*.



A BIPOLAR STEPPER MOTOR CIRCUIT

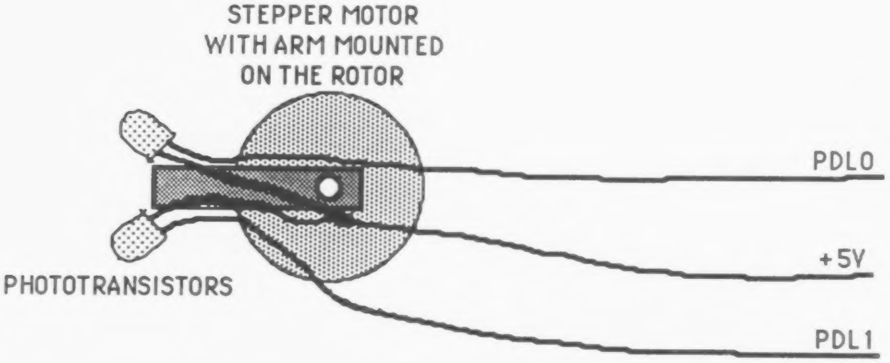
To control a bipolar stepper motor with this circuit, the switches must be operated in a different pattern. The program STEPPER.B on the *Project Programs* disk controls the Optically-Isolated Switches in the proper order for a bipolar stepper motor. Bipolar stepper motors can also be operated using transistorized-switching circuits and IC stepper-motor controllers.

◊ Use an IC stepper-motor controller

The standard way to operate a stepper motor is with an IC specifically designed to do the switching needed to cause the proper motion. These ICs are available from some of the same companies that sell stepper motors. A good article on interfacing stepper motors with IC controllers using an Apple computer is in the October, 1983 issue of *InCider*, p. 122. One advantage of using an IC controller is that you need only two control lines from the computer for each controller. This means you could control two stepper motors from the Optically-Isolated Switches.

◊ Add light sensors to your stepper motor

One interesting experiment that can be done with the stepper motor is to use it as a "sun seeker." Mount two phototransistors on a short arm that is attached to the rotor of a stepper motor. The orientation and the electrical connections of the two light sensors are shown below:



By using the PDL(0) and PDL(1) commands, a BASIC program can compare the light intensity in the directions that the two phototransistors are pointing. In general, one phototransistor or the other will be receiving more light. Program the computer to compare the two light intensities and then move the rotor toward the brighter area. If programmed properly, the rotor arm should always point toward the brightest light source. The book *Apple II Applications* by Marvin L. DeJong explains how to use this method to make a practical solar tracker, useful for aiming a solar collector.

◊ Turn your stepper motor into a part of a robot

Stepper motors are ideal for use in robotics, because of the precision with which they can be controlled. To do much in robotics, you will probably want to use more than just one stepper motor. If you use IC stepper-motor controllers (the second project extension listed above) you can control two stepper motors from the game port. If you need to use more, you will need to use an interfacing board or some other, more complicated hardware.

Program Improvements

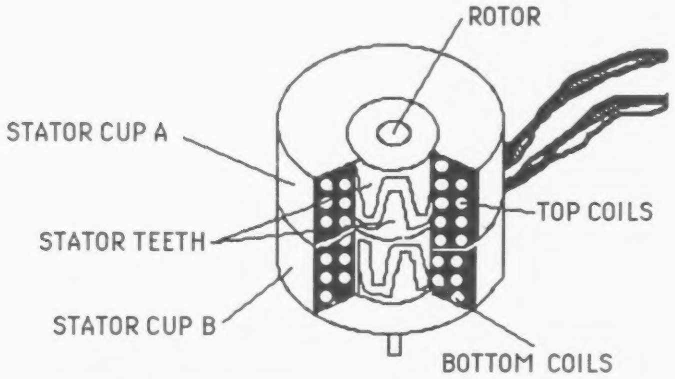
◊ Try speeding up the stepper-motor rotation

The speed of the stepper-motor rotation is limited by the BASIC program STEPPER. You may be able to speed up the motion by "streamlining" the program. Another way to speed up the motion is to compile the BASIC program. The best way to speed things up is to rewrite the STEPPER program in machine language. Most stepper motors are capable of rotating at a rate of several hundred steps per second.

SUPPLEMENTAL INFORMATION

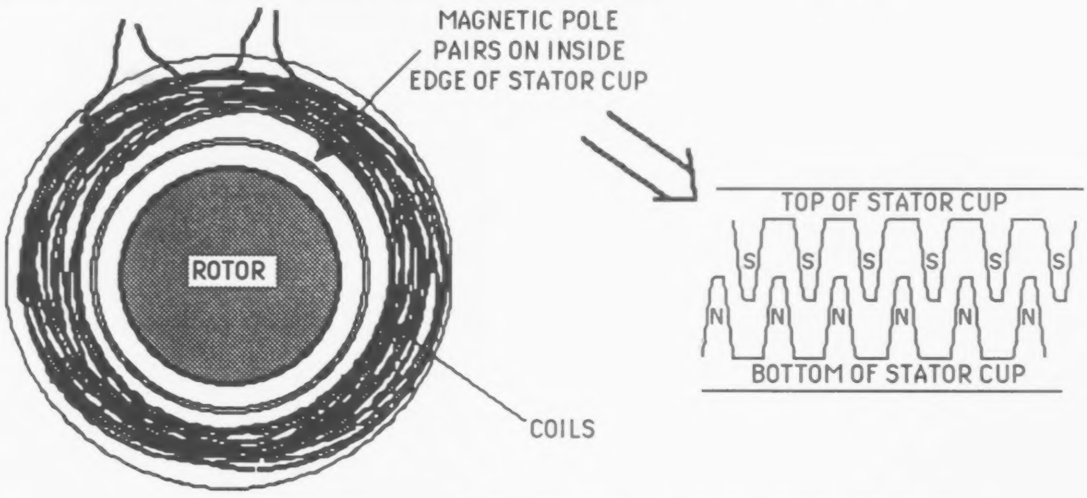
Real Stepper Motors

Real stepper motors are more complex than the hypothetical motor explained in the "Background Information" section above. The inside of a stepper motor consists of two donut-shaped *stator cups*. These are iron pieces that surround the coils. The two stator cups are stacked on top of each other and the rotor is in the center.


ROTOR
STATOR CUP A
TOP COILS
STATOR TEETH
STATOR CUP B
BOTTOM COILS

CUTAWAY DIAGRAM OF A STEPPER MOTOR

The stator cups become magnetized when the coils are turned on. Each stator cup has a series of "teeth" on its inside edge, which form magnetic-pole pairs.

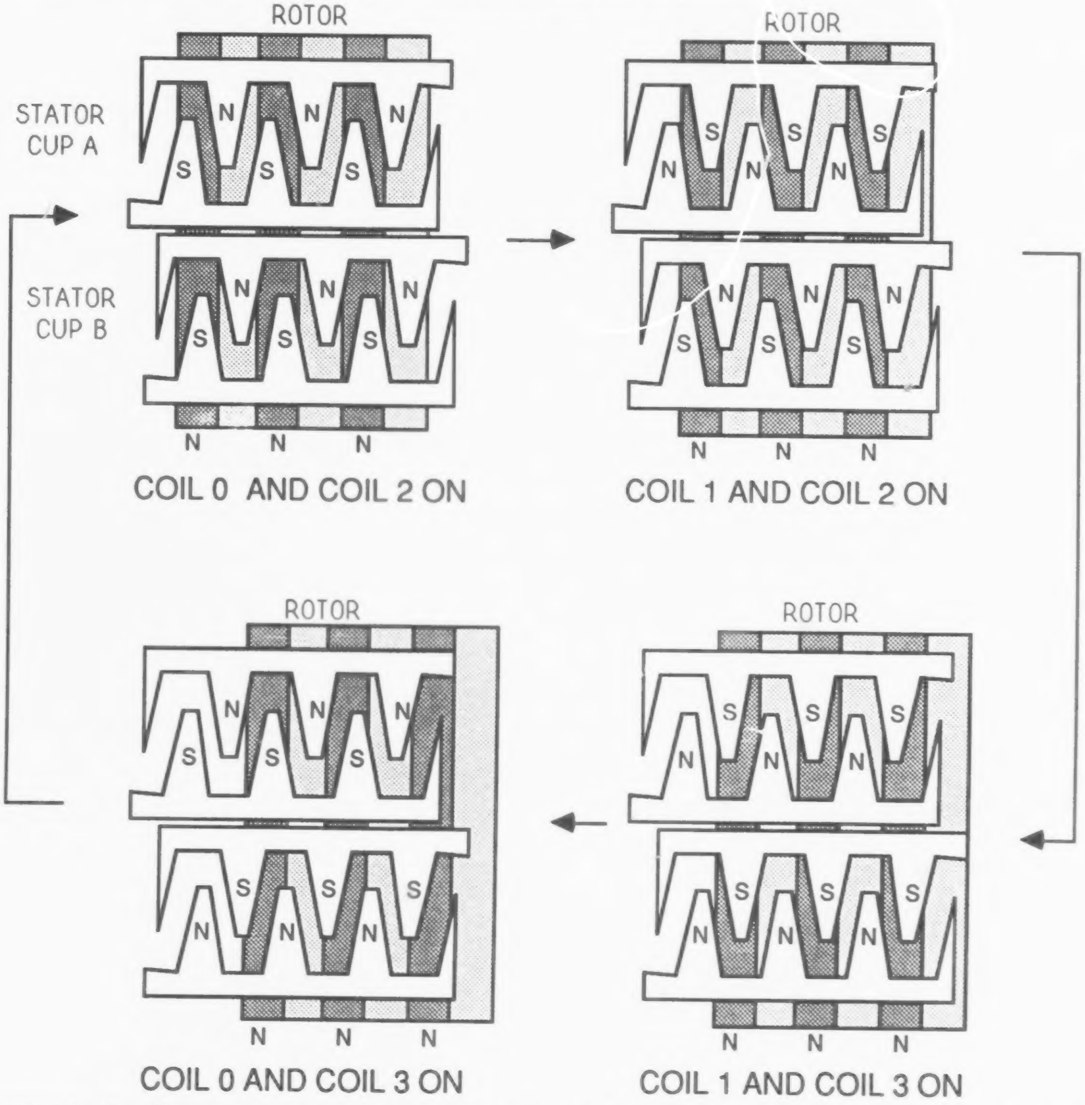

MAGNETIC POLE PAIRS ON INSIDE EDGE OF STATOR CUP
ROTOR
TOP OF STATOR CUP
S
S
S
S
S
S
N
N
N
N
N
N
BOTTOM OF STATOR CUP
COILS

TOP VIEW OF STEPPER MOTOR SHOWING STATOR CUP TEETH

In a unipolar stepper motor, there are two coils inside each stator cup. Current may flow in one or the other of these coils (never both at the same time) to magnetize the stator. Depending on which of the coils is "on," either the top or bottom of the teeth on the stator cup will be magnetic north poles.[6] The rotor is a permanent magnet with several pairs of north and south poles along its outside edge. By controlling the coils, the rotor can be attracted to different positions. The diagram below illustrates the motion of the rotor in the Normal drive method. Coils 0 and 1 are in stator cup A; coils 2 and 3 are in stator cup B. Using this drive method, one of the coils in each stator cup is always on. The rotor positions itself according to the magnetic fields of both sets of stator teeth.

---

[6] The stators of a *bipolar* stepper motor are very similar to the stators of a unipolar stepper motor. The difference is in the coil wiring. In a bipolar stepper motor there is only one coil of wire in each stator cup. A bipolar motor requires circuitry to allow the current in the stator coil to flow in either direction.



COIL 0 AND COIL 2 ON

COIL 1 AND COIL 2 ON

COIL 0 AND COIL 3 ON

COIL 1 AND COIL 3 ON

The rotor in the diagram is shown moving in a line rather than rotating. Moving around the four diagrams in a clockwise direction corresponds to the rotor moving clockwise. Moving around the diagrams in a counterclockwise direction corresponds to the rotor moving counterclockwise.

When the Wave drive method is used, only one of the four coils in the motor is on at a time. The rotor positions itself according to the magnetic field of the one set of stator teeth that is magnetized.

The Half-Step drive method combines the Normal and Wave drive methods to provide twice as many possible rotor positions.[7]

---

[7] For more information on how stepper motors work, refer to the "Stepper Motor Handbook" from Airpax Corporation, Form MR 116-R3, 1982.

Stepper-Motor Specifications

The following are terms used in describing stepper motors:

Step Angle: The angle that the rotor moves for each normal step.

Step Accuracy: The amount a particular step differs from the nominal step angle. It can be expressed as a percent or in degrees.

Holding Torque: The torque required to rotate the stepper-motor rotor when the current is on, holding the rotor on a particular step.[8]

Dynamic Torque: The torque that the motor can deliver in rotation. This torque varies with the speed of the motor.

Residual Torque: The torque required to rotate the stepper-motor rotor when there is no current. This torque is caused by the small amount of magnetism that remains in the stator when the current is off.

Program Design Notes

Each of the three drive methods requires a different pattern of switch settings. The Normal drive method involves a four-step sequence:

CLOCKWISE ROTATION ↓

STEP	S0	S1	S2	S3
1	ON	OFF	ON	OFF
2	ON	OFF	OFF	ON
3	OFF	ON	OFF	ON
4	OFF	ON	ON	OFF

↑ COUNTERCLOCKWISE ROTATION

Notice that two coils are turned on at a time. Each step through the pattern rotates the rotor one step angle.

You can also cause the rotor to stop at the midpoints between the standard steps by using the following pattern:

CLOCKWISE ROTATION ↑

STEP	S0	S1	S2	S3
1	ON	OFF	ON	OFF
2	ON	OFF	OFF	OFF
3	ON	OFF	OFF	ON
4	OFF	OFF	OFF	ON
5	OFF	ON	OFF	ON
6	OFF	ON	OFF	OFF
7	OFF	ON	ON	OFF
8	OFF	OFF	ON	OFF

↓ COUNTERCLOCKWISE ROTATION

This is called the Half-Step sequence. Only one coil is turned on every other step. Each step in the sequence advances the rotor 1/2 step angle. The disadvantage of this drive method is that the holding torque of the motor is only half as strong on every other step.

---

[8] In physics, *torque* is defined as the force applied multiplied by the perpendicular component of the lever arm through which it acts. Typical units for describing the torques of small stepper motors are millinewton-meters or ounce-inches.

A third drive method is called Wave drive. The sequence of steps is shown below:

CLOCKWISE ROTATION ↑

STEP	S0	S1	S2	S3
1	ON	OFF	OFF	OFF
2	OFF	OFF	OFF	ON
3	OFF	ON	OFF	OFF
4	OFF	OFF	ON	OFF

COUNTERCLOCKWISE ROTATION ↓

Only one coil is used at a time in this sequence. The advantage is reduced power consumption. The disadvantages are decreased power and step accuracy.

These three drive methods are all possible using the BASIC program STEPPER. When the program is first run, you are asked to select which of the three drive methods you wish to use. Next you enter the number of steps you want the rotor to rotate. Use positive numbers for clockwise rotations and negative numbers for counterclockwise rotations.

The actual switch control is then executed by the program using the BASIC statements listed below:

```
100 POKE C0, 0 : POKE N1,0 : POKE C2,0 : POKE N3,0 : REM S0 AND S2 ON
200 POKE C0, 0 : POKE N1,0 : POKE N2,0 : POKE N3,0 : REM S0 ON
300 POKE C0, 0 : POKE N1,0 : POKE N2,0 : POKE C3,0 : REM S0 AND S3 ON
400 POKE N0, 0 : POKE N1,0 : POKE N2,0 : POKE C3,0 : REM S3 ON
500 POKE N0, 0 : POKE C1,0 : POKE N2,0 : POKE C3,0 : REM S1 AND S3 ON
600 POKE N0, 0 : POKE C1,0 : POKE N2,0 : POKE N3,0 : REM S1 ON
700 POKE N0, 0 : POKE C1,0 : POKE C2,0 : POKE N3,0 : REM S1 AND S2 ON
800 POKE N0, 0 : POKE N1,0 : POKE C2,0 : POKE N3,0 : REM S2 ON
```

These eight statements are the commands necessary to rotate the rotor using the Half-Step sequence. To move the rotor clockwise in the Half-Step sequence, the program proceeds downward through these statements for as many statements as there are steps to execute. To move the rotor counterclockwise in the Half-Step sequence, the program proceeds through the statements in reverse order.

The same BASIC statements can be used for the Normal drive sequence. The Normal drive sequence consists of every other statement in the list above (statements 100, 300, 500 and 700). To rotate the rotor clockwise using the Normal drive method, the program advances through the statements down the list, using only every other statement. Normal drive method counterclockwise rotation is done by moving up the list, skipping every other statement.

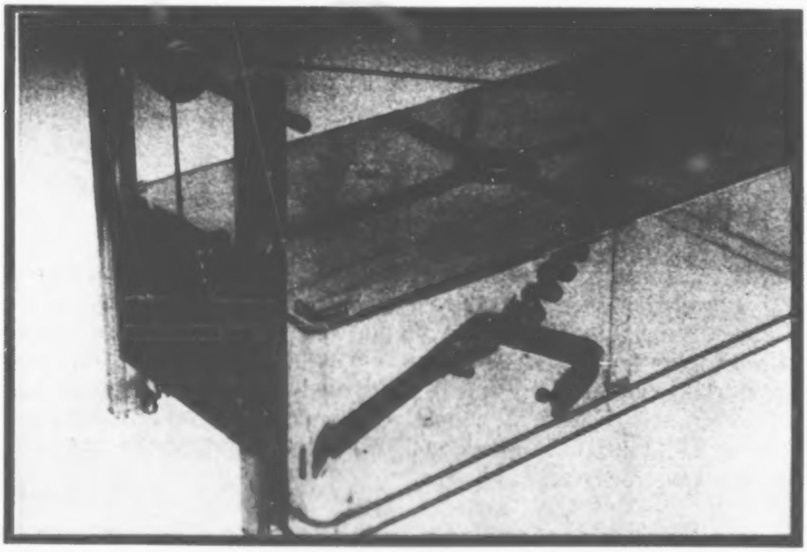
Wave drive is done in the same way as Normal drive, except that only statements 200, 400, 600 and 800 are used.

The program STEPPER on the *Project Programs* disk includes the line:

```
900 POKE N0, 0 : POKE N1,0 : POKE N2,0 : POKE N3,0 : RETURN : REM ALL
SWITCHES OFF
```

Each time the program finishes executing a series of steps, a GOSUB 900 statement is used to turn off all electricity flowing to the stepper motor. This step was added to save energy, which is important when you are operating the stepper motors from batteries. However, turning off the current means that the rotor will not be held in place (except by residual torque). If you want holding torque while the program waits for you to enter a new command, delete line 47 of the program.

PROJECT 14:
A BETTER MOUSETRAP



In this project you will construct a "high tech" mousetrap by combining three of the previous projects in this book. A photogate (Project 1) is used to detect the mouse in the trap. The Optically-Isolated Switches (Project 11) are used to control a stepper motor (Project 13) to lower a trap door and catch the mouse. A testimonial to the success of this device is that it caught fourteen mice at Vernier Software in one week!

PROJECT REQUIREMENTS

Project Prerequisites

The photogate from Project 1, the Optically-Isolated Switches from Project 11 and the stepper motor from Project 13 are all required for this project.

Background Knowledge

- General: Building the mousetrap device requires some construction skills.
- Electronics: None. If you have done Projects 1, 11 and 13, this is an easy project.
- Programming: Some experience with BASIC is recommended. The core project involves changing one line of a BASIC program and then saving the modified program.

Parts

☐ plastic shoebox or metal container that can be used as a cage in which to trap the mouse (Plastic shoeboxes are available at discount or variety stores. We recommend a clear, plastic shoebox. This will let you see what is going on inside the cage.)

☐ metal plate to use as a trap door

☐ (2) "U-shaped" plastic or metal pieces for mounting trap door

☐ pulley to mount on stepper-motor shaft (a sewing-machine bobbin can be glued to the shaft for this)

- ☐ pulley that can be mounted to rotate freely (a sewing-machine bobbin can also be used for this)
- ☐ string
- ☐ miscellaneous odds and ends for mounting the free-wheeling pulley and the stepper motor

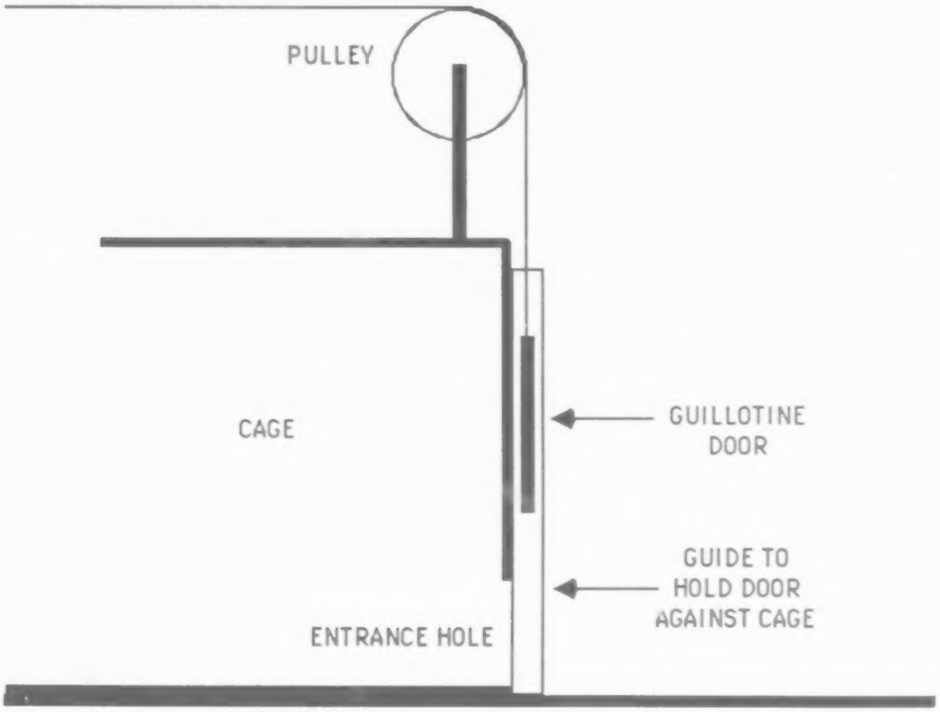
Tools and Supplies

Solder, soldering iron, glue, pliers, drill, drill bits, screwdriver, wire strippers, and cheese or peanut butter. You will also need a DC power supply to power the stepper motor. You can use the same power supply you used in Project 13.

BACKGROUND INFORMATION

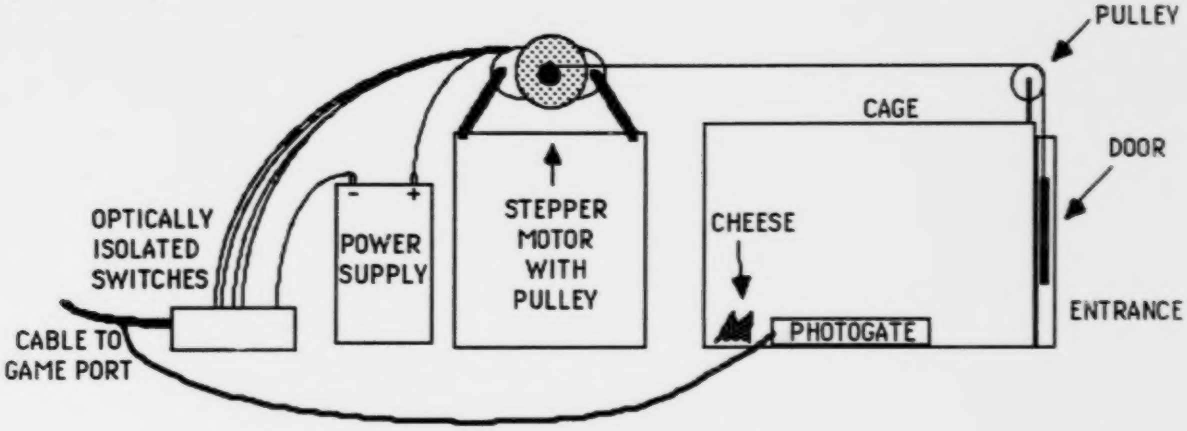
A Better Mousetrap is an example of putting several of the ideas in this book together for some new experiments. For this project you need to construct a mouseproof cage with a single, small opening. The mouse will hopefully be lured into the cage and then trapped electronically. The electronic mousetrap uses a photogate as a means of detecting when a mouse has entered the cage. This photogate is identical to the one used in Project 1. It consists of an LED, which produces a infrared light beam, and a phototransistor, which changes resistance when the infrared light beam strikes it. The photogate is connected to the PB2 input signal on the game port, so the computer can sense when the photogate beam has been interrupted.

The entrance to the cage should have a "guillotine" trap door mounted over it that can move down and prevent the mouse from leaving the cage. The door is suspended from a string hung over a pulley mounted at the top of the cage.



The string holding up the guillotine door is connected to a pulley fastened onto the shaft of a stepper motor. By controlling the rotation of the stepper-motor rotor, the string is reeled in or out, moving the door up or down.

The program MOUSETRAP on the *Project Programs* disk can be used to control the system. When everything is set up, the program is run. It constantly checks the status of the photogate, waiting for the first time the gate is blocked. When (or if) the photogate is blocked by a mouse, the program detects the change in the voltage on the PB2 input line and begins executing a sequence of Optically-Isolated Switch settings. This causes the stepper motor to rotate in the correct direction and for the proper distance to lower the door, capturing the mouse. The mouse is uninjured. The complete system is diagrammed below:



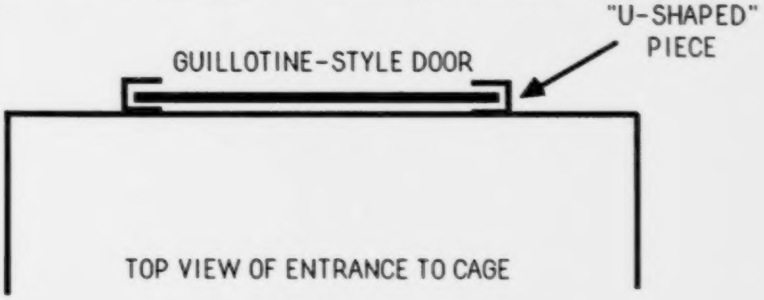
A BETTER MOUSETRAP

Since stepper motors are relatively slow, you may be surprised that the mouse does not just quickly run out of the door before it closes. This has never happened in our tests. Apparently, the mouse "freezes" when it hears (or feels) the door start to move.

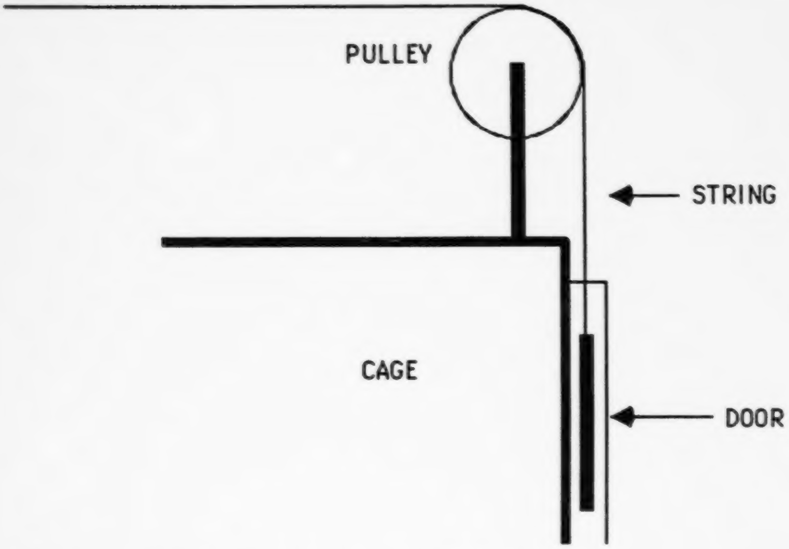
STEP-BY-STEP INSTRUCTIONS

☐ 1. **Construct the mousetrap cage.** Use a container that is tough enough so a mouse cannot chew through it easily. A clear, plastic shoebox is recommended; it will allow you to see what is going on inside the cage. The top of the shoebox can be used as the "floor" of the cage. This will allow you to transport the mouse after you catch it. Metal hamster cages and birdcages probably will not work because mice can squeeze through the small spaces between bars.

You will need a small hole at ground level to use as the entrance. Select a flat, metal plate to use as a trap door. Glue or bolt two "U-shaped" metal or plastic strips vertically on either side of the hole. These pieces should hold the door against the cage, while still allowing it to move up and down freely. The diagram below shows a top view of the cage with the trap door holder in place.



Mount a pulley on the top of the cage so it can be used as shown below:



The pulley should be able to turn freely so the trap door will lower easily. A sewing-machine bobbin, slipped over a narrow rod, can be used as this pulley.

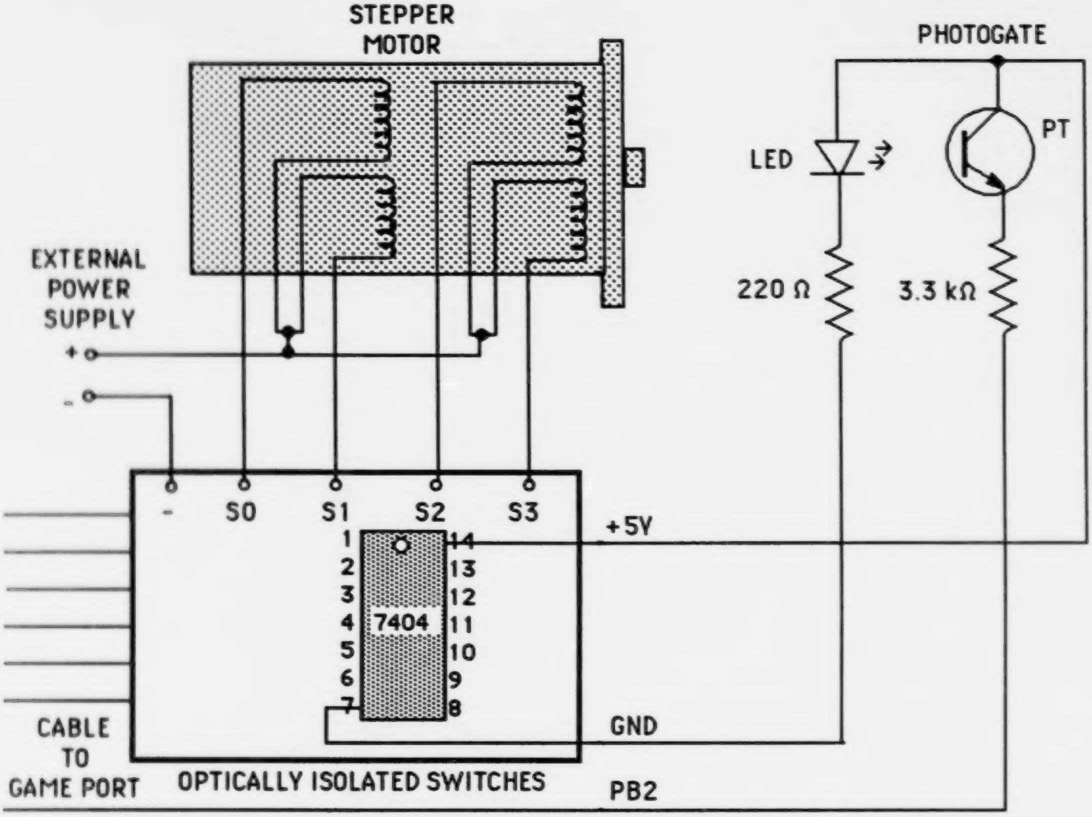
☐ 2. **Connect the mousetrap door to the stepper motor.** Glue a pulley rigidly to the shaft of the stepper motor. A sewing-machine bobbin can probably be used for this. Tie a section of string to this pulley. Wrap the string around the pulley several times and then connect it to the top of the trap door. Anchor the stepper motor to a block of wood or some other heavy object. Arrange the apparatus so that when the stepper-motor rotor rotates, the trap door will move up or down.[1]

☐ 3. **Connect the leads of the stepper motor to the Optically-Isolated Switches and to the power supply.** The connections should be the same as you used in Project 13. Refer to the instructions in that project, if necessary.

☐ 4. **Connect the photogate circuit to the Apple game-port cable.** The photogate must be connected to the same cable that is used to connect the Optically-Isolated Switches to the game port. Connections must be made to the +5V, PB2 and GND leads. The PB2 lead is not used on the Optically-Isolated Switches, so you should be able to solder onto it easily. The +5V line is connected to pin #14 of the 7404 IC in the switch. The GND line is connected to pin #7 of the 7404 IC in the switch. The complete wiring diagram for this project is shown below:

---

[1] A stepper motor with a gear-reduction system mounted on it may not be good for this project, since it would rotate the pulley very slowly. If the trap door is lowered too slowly, the mouse may escape. In our (successful) trapping tests, the door was lowered in about seven seconds. The gear-reduction system can usually be removed from the stepper motor.



WORKING DIAGRAM OF A BETTER MOUSETRAP

☐ 5. **Test the photogate.** With the computer off, connect the Optically-Isolated Switches and the photogate to the game port. Turn on the computer and run the MOUSETRAP program. This program includes two modes specifically designed for testing. First, select mode P (TEST PHOTOGATE). The status of the photogate will be displayed on the monitor. It should be "HIGH" when the photogate is unblocked. Move your finger through the gate. The display should be "LOW" when the gate is blocked. If the photogate does not work properly, refer to the "Troubleshooting" section below. To return to the main menu, press <RETURN>.

☐ 6. **Test the trap door.** Make sure the external power supply is turned on. Select mode D (DOOR TEST). Adjust the string and the stepper motor so the string is tightly stretched and the door is all the way down. When you enter "U" for up, the stepper motor should rotate and raise the door high enough to allow a mouse to enter. If the rotation of the rotor is the wrong way, wrap the string the other way around the stepper-motor pulley. If the door does not raise high enough or raises too high, exit the program by pressing <CONTROL> <C>. To adjust the door movement, you need to change one line of the MOUSETRAP program. List the program and change the value of N in line 51. This is the number of steps the rotor rotates in raising the door. Increase N if you want the door to raise higher; decrease N if you want less motion. Run the program again and retest the door. When you have the motion of the door adjusted properly, save the MOUSETRAP program on the disk with your changes made in line 51.

☐ 7. **Begin trapping.** Place the photogate inside the cage. Put some food that will be appetizing to the mouse[2] in a position such that the mouse will have to move through the photogate to reach it. Set up the door apparatus with the door all the way down and the string tight. Select mode T (TRAP) from the main menu. The program will first raise the trap door to the proper height, and then begin continuously checking the photogate.[3] When the photogate is blocked, the program controls the Optically-Isolated Switches so that the stepper motor will lower the trap door. If you want to exit this routine before you catch a mouse, press <Q>. Good luck with your trapping!

TROUBLESHOOTING

If you have problems with this project, first check for the likely errors:

- Make sure the Optically-Isolated Switches are connected correctly and working properly. Connect a simple test circuit to the switch unit and test it with the SWITCH program.
- Check for a wiring error. It is easy to confuse the leads from the stepper motor. Also, make sure the photogate is connected to the proper leads on the game-port cable.
- Make sure the power supply is connected correctly and is "on." Use a voltmeter, if necessary, to make sure the proper voltage (5 or 12 volts, depending on the stepper motor being used) is being supplied.

If none of these problems are detected, refer to the "Troubleshooting" section of either Project 1 (for problems in the photogate circuit) or Project 13 (for problems with the stepper motor).

PROJECT EXTENSIONS

Program Modifications

◊ Turn off the current to the stepper motor while waiting for the mouse

Mode T of the MOUSETRAP program normally raises the door and then leaves two of the stepper-motor coils on to hold the door up. This takes quite a lot of current. If you are using batteries as your power supply, the batteries may go dead before you ever catch a mouse.[4] You may be able to save energy by turning off the stepper-motor coils while you are waiting for the mouse. You can only do this if there is enough ***residual torque*** (holding torque with the power off) to keep the door from falling down when you turn off the current. Try adding the following line to the MOUSETRAP program.

```
4045  GOSUB 900 :  REM TURN OFF ALL SWITCHES
```

Run the program and select mode T. Raise the door into position for trapping. Check to see if the door stays up. If it does, you can use the program with this line added and conserve energy.

◊ Change drive modes

The MOUSETRAP program uses only the Normal drive method (Half-Step or Wave) described in Project 13. You can change it to use one of the other drive methods. Use the STEPPER program on the *Project Programs* disk as a model.

---

[2] We have had success with the old favorite—cheese. A peanut butter topping on cheese is also recommended.

[3] As written, the MOUSETRAP program will leave two of the electromagnets in the stepper motor on to hold the rotor in position while it waits for the mouse. This holds the trap door up, but it takes quite a lot of energy. One of the "Project Extensions" involves turning off the stepper motor coils during this part of the program to save energy.

[4] This is not a serious problem if you have a lot of energy or a lot of mice.

Circuit Modifications

◊ Add a second photogate to the system

Wire a second photogate like the original photogate, except connect the "signal" lead to the PB0 input line instead of the PB2 line.[5] This second gate can be used to indicate the position of the mouse inside the cage. Perhaps you can design a system capable of catching two mice in one session. Modify the MOUSETRAP program to respond to changes in the status of the PB0 line.

◊ Experiments in animal behavior

With only minor modification, the hardware and software from this project can be used for many animal behavior experiments. The stepper motor can release food for an animal whenever a specific procedure is performed. The photogate (or photogates) can be used to detect the actions of the animal. For example, hamster food can be released each time a hamster blocks the photogate. Experiments can be done on how animals learn.[6]

SUPPLEMENTAL INFORMATION

Program Design Notes

The MOUSETRAP program is made up of several distinct sections, each of which is explained below:

Lines 3- 45: Initialization
These lines define the constants used to turn on and off the Optically-Isolated Switches and present the main menu.

Lines 48-87: Driver Subroutine
This is a subroutine used to cause the rotor to rotate N steps in the direction defined by the variable D. The number of steps, N, is specified in line 51. This value may be changed to increase or decrease the distance the door moves. The direction of the door's motion is up if the value of D is +1 when the subroutine is entered. It is down if the value of D is -1.

Lines 100-900: Driver Steps
These are the statements that actually turn on or off the Optically-Isolated Switches. These statements are used by the Driver Subroutine. Line 900 is a statement that turns off all four switches to save energy. It is used several times in the program. The program uses only the Normal drive method.

Lines 2000-2080: Photogate Test
This short section tests the status of the PB2 input line. It displays "LOW" if the photogate is blocked and "HIGH" otherwise.

Lines 3000-3400: Door Test
This mode allows you to raise or lower the trap door manually. It allows you to make sure the door is working properly before you begin trapping.

Lines 4000-4500: Trap
This part of the program first raises the trap door. It then continually checks the status of the photogate by using a PEEK statement. If it detects that the photogate is blocked, it lowers the trap door. While waiting for a signal from the photogate, the program also checks for a keypress. If any key is pressed, the program returns to the main menu.

---

[5] We recommend avoiding the use of the PB1 line because of the Apple "self-test" problem on the Apple IIc. This is discussed in "The Apple Game-Port" section of the *Introduction*.

[6] One example of animal behavior experiments is "The Pigeon Perception of Letters of the Alphabet" by Donald S. Blough in *Science*, October 22, 1982, p. 397.

Questions and Answers about *A Better Mousetrap*

Q. *If I build A Better Mousetrap, will the world beat a path to my door?*

A. That's what they always say. We do not know for sure.

Q. *Where should we purchase the electronic components for A Better Mousetrap?*

A. Mouser Electronics is recommended for this project.

Q. *How difficult is the Better Mousetrap project?*

A. If you have completed Projects 1, 11 and 13 of the book, this is a real Mickey Mouse project.

Q. *Can the electronic mousetrap be used to capture moles?*

A. That is a very good question. As you probably know, the light source (an LED) in the photogate of the mousetrap produces mostly infrared light.[7] This means that it is very difficult for humans to detect any light coming from the LED. In total darkness, however, there is a faint red glow from the LED. Now if you mount your mousetrap completely underground (where moles hang out), it will obviously be very dark. We do not know for sure what will happen. Will the mole notice the faint red glow of the LED or are moles blind? Do they have eyes that are more or less sensitive to infrared light than human eyes? If they do see it, will they be frightened away or attracted? As you can see, there are many good topics for further research here. Dig in!

Q. *Can the computerized mousetrap be run on a single disk-drive system with only a monochromatic video monitor?*

A. We feel that a single disk-drive and a monochromatic monitor may be entirely appropriate. After all, with the extra disk drive and a color monitor, the cost of the computer hardware alone would be nearly $2,000.00. That works out to quite a lot per mouse, even if you use the computerized mouse-trap a lot.[8] Sometimes, one has to be practical.

Q. *What if our house doesn't have mice?*

A. Consider yourself lucky. Practice on a gerbil.

Q. *Why should I build A Better Mousetrap when a standard mechanical mousetrap costs only $1.45?*

A. A Better Mousetrap is more humane, less messy, and quieter. Besides, how much electronics or computer science can you learn using a mechanical mousetrap?

---

[7] Its peak emission is at a wavelength of 880 nm. Human eyes do not detect much light of wavelength longer than 700 nm.

[8] In most situations, you can probably catch only one mouse per night (unless you modify the hardware and software as described in the project extensions). That is only 365 mice per year, so if you use the system for 5 years straight, you could catch 1825 mice. Even at that, it is about a dollar a mouse just for the computer hardware, not to mention your time in constructing and operating the system.

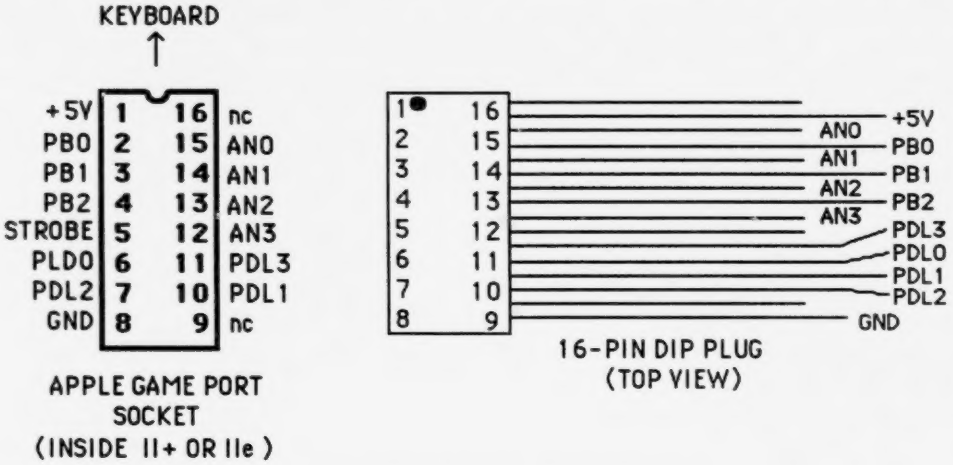
Appendix A

Cables to the Apple Game Ports

If you are using an Apple II+ you need to use a 16-conductor ribbon cable to the internal game port. If you are using an Apple IIe, you have a choice of two different cables to use: a 16-conductor cable to the internal game port or a 9-conductor cable to the external game port. These two types of cables are described below:

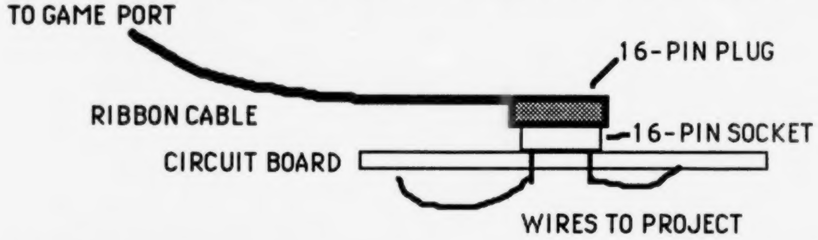
16-Conductor Cables to the Internal Game Port (II+ or IIe)

The Apple II+ and IIe both have a 16-pin DIP (Dual In-line Pin) connector inside the computer for use as a game port. To use this connector, you have to remove the top cover of the computer. The connector is in the right rear portion of the computer and looks like an IC socket. The pins and a ribbon cable with a plug to connect to the connector are shown below. Notice that there is a notch on the end of the connector nearest the keyboard. This notch indicates pin #1 of the game port.


KEYBOARD
+5V 1 16 nc
PB0 2 15 AN0
PB1 3 14 AN1
PB2 4 13 AN2
STROBE 5 12 AN3
PLD0 6 11 PDL3
PDL2 7 10 PDL1
GND 8 9 nc
APPLE GAME PORT SOCKET (INSIDE II+ OR IIe)
1 16
2 15
3 14
4 13
5 12
6 11
7 10
8 9
AN0
AN1
AN2
AN3
+5V
PB0
PB1
PB2
PDL3
PDL0
PDL1
PDL2
GND
16-PIN DIP PLUG (TOP VIEW)

APPLE INTERNAL GAME PORT

16-conductor ribbon cables with DIP plugs are sold at most electronic-parts supply houses. If available, purchase a 16-conductor ribbon cable with a DIP plug on only one end. You can solder the appropriate leads on the other end to your project. Another alternative is to purchase a cable with a DIP plug on each end. Radio Shack sells a short, 45-cm (18-inch) cable with a plug on each end (RS #276-1976). There are two ways to use a cable such as this. You can cut one of the plugs off and solder the appropriate leads to your project. If you want to be able to use the same cable for other projects, you can insert the second plug into a 16-pin DIP IC socket (RS #276-1998) mounted on your project. Connect the wires from the pins of this socket to the project as shown below:


TO GAME PORT
16-PIN PLUG
RIBBON CABLE
16-PIN SOCKET
CIRCUIT BOARD
WIRES TO PROJECT

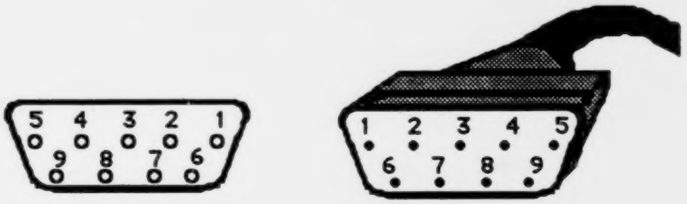
When using a 16-conductor ribbon cable, mark one corner of the plug as the #1 pin and always match this pin with the notch on the connector. It is easy to plug a 16-pin DIP plug in the wrong way. Another potential problem with this kind of connector is bending the pins. The pins are made of very fine wire and can easily bend as you plug them in. Once they bend, they often break when you straighten them out. Worse yet, sometimes you do not notice that the pins have bent instead of making connection, and your circuit will not work. To avoid these problems, make sure all of the pins are straight and in line as you plug in the connector.

There are devices sold that act as "extension cords" for the 16-pin game port and move the point of connection outside the computer.[1] These "game-port extenders" can be real time savers because they allow you to connect to the internal game port without taking the top off the computer. This kind of apparatus is probably a good investment if you are going to use the 16-pin Apple game port frequently.[2]

9-Conductor Cables to the External Game Port (IIe Only)

The Apple IIe has a second game port located on the outside back of the computer. It is somewhat more convenient to use, because you do not have to take the top off the computer to reach it. This game port uses a 9-connector sub-D socket. The annunciator-output lines AN0, AN1, AN2, and AN3 are not present on this connector. This means the 9-conductor cable **cannot** be used for any project that involves controlling electrical devices. Specifically, this connector cannot be used for the following projects: *Optically-Isolated Switches*, *Computer-Controlled Car*, *Stepper Motor*, and *A Better Mousetrap.*

1 = PB1
2 = +5V
3 = GND
4 = PDL2
5 = PDL0
6 = PB2
7 = PB0
8 = PDL1
9 = PDL3



GAME PORT SOCKET
(OUTSIDE BACK OF IIe)

9-PIN SUB D PLUG
(WITH PINS POINTING
OUT OF PAGE AT YOU)

APPLE IIe EXTERNAL GAME PORT

The plug that goes to this game port is a 9-conductor sub-D plug, which is available at Radio Shack (RS #276-1537) and most other electronic-parts supply houses. Solder either a ribbon cable or a multiple-conductor wire to this connector. If you do not want to spend time soldering leads onto the plug, purchase a Radio Shack Joystick Extension Cable (RS #276-1978). This is a 10-foot long cable with a sub-D plug (pins sticking out) on one end and a sub-D socket (with holes) on the other end. You can use the plug end to connect to the Apple IIe. Cut the other end off and throw it away.[3] Strip the wires on the cut end of the plug cable for soldering to your project. If you use the Radio Shack Joystick Extension Cable, the leads are identified by color as follows:

1 = PB 1 (green)
2 = +5V (yellow)
3 = GND (orange)
4 = PDL2 (red)
5 = PDL0 (brown)
6 = PB 2 (blue)
7 = PB 0 (black)
8 = PDL1 (gray)
9 = PDL3 (white)

---

[1] Two such devices are: (1) Paddle-Adapple from SCRG, P.O. Box 593, Moorpark, CA 93020, (800) 635-8310, and (2) Game Socket Extender from Happ Electronics, 4640 Island View, Oshkosh, WI 54901, (414) 231-5128.

[2] Many of these devices also have a *zero insertion force socket* on one end. This is a special kind of 16-pin socket that lets you slip the 16-pin DIP plug into the socket with almost no friction. You then move a lever that clamps down on the pins. This greatly reduces the chances of bending and breaking pins.

[3] The end that is discarded has a 9-conductor sub-D socket on it. This is the connector used on Atari 2600 video games or Commodore-64 computer control ports, so you may find a use for this end of the cable.

Appendix B

Resistor Color Codes

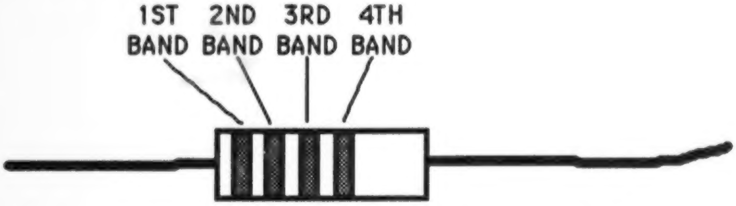
Almost all resistors are labeled using the following colors to represent numbers:

black	0	blue	6
brown	1	violet	7
red	2	gray	8
orange	3	white	9
yellow	4	gold	0.1
green	5	silver	0.01

The labeling system used is slightly different for carbon-composition resistors and metal-film resistors.

Carbon-Composition Resistors

Most ordinary carbon-composition resistors are labeled with four color bands. The bands are read in order starting at the end of the resistor with a color band nearest it. The figure below illustrates this:



The first two bands are read as a two-digit number. The third band is read as a power of 10 multiplier of that number.

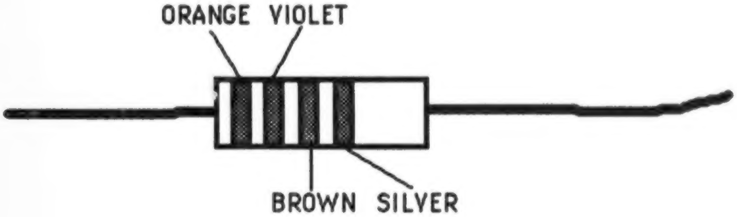
$$\text{resistance value} = (\text{2-digit \# from first 2 bands}) \times 10^{(\text{third band digit})}$$

The fourth band is an indicator of the tolerance of the resistor. The tolerance is stated as a percent of the labeled value, which is the most that the actual resistance can differ from the labeled value. A special color code is used for the tolerance color code:

no color	20%	gold	5%
silver	10%	brown	1%

Consider some examples:

Example 1:

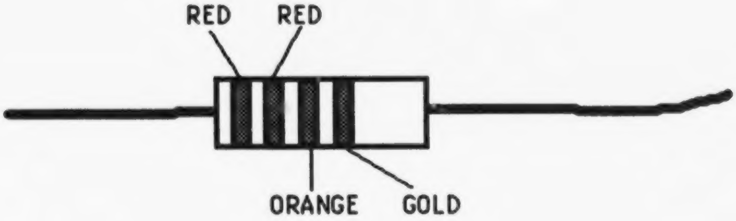


The value is:

$$37 \times 10^{1} = 37 \times 10 = 370\ \Omega$$

The tolerance is plus or minus 10%.

Example 2:



The value is:

$$22 \times 10^3 = 22 \times 1000 = 22{,}000\ \Omega \text{ or } 22\ \text{k}\Omega$$

The tolerance is plus or minus 5%.

Metal-Film (Precision) Resistors

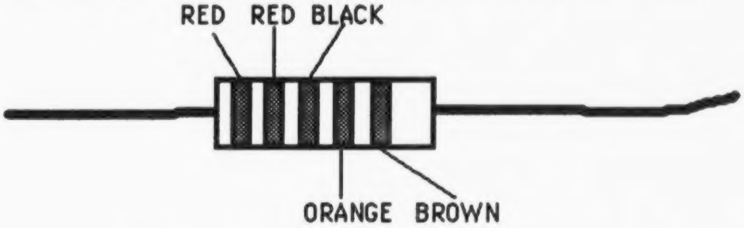
Metal-film resistors are labeled slightly differently. They are marked with five color bands. The first three indicate a three digit number. The fourth band is a power of 10 multiplier of that number.

$$\text{resistance value} = (\text{3-digit \# from first 3 bands}) \times 10^{(\text{fourth band digit})}$$

The fifth band is used to represent the tolerance of the resistor. It is usually brown because most metal-film resistors have a tolerance of 1%.

Sometimes it is difficult to determine which end of the resistor to begin with when using metal-film resistors. Remember that the fifth band on a metal-film resistor is usually brown. If all else fails, you can check the resistance with an ohmmeter.

Here is an example of reading a metal-film resistor's value from the color bands:



METAL-FILM PRECISION RESISTOR

The value is:

$$220 \times 10^3 = 220 \times 1000 = 220{,}000\ \Omega \text{ or } 220\ \text{k}\Omega$$

The tolerance is plus or minus 1%.

Appendix C

Bibliography

Electronics

American Radio Relay League. *The Radio Amateur's Handbook.* Newington, CT: 1985.

Carr, Joseph J. *The Complete Handbook of Amplifiers, Oscillators and Multivibrators* . Blue Ridge Summit, PA: TAB Books, 1981.

Horowitz, Paul and Winfield Hill. *The Art of Electronics*. Cambridge: Cambridge University Press, 1980.

Kuecken, John A. *How to Measure Anything with Electronic Instruments*. Blue Ridge Summit, Pennsylvania: TAB Books, 1981.

Malmstadt, Howard V., C.G. Enke and S.B. Crouch. *Electronics and Instrumentation for Scientists* . New York: Benjamin-Cummings, 1982.

Meiksin, Z. H. and Philip C. Thackray. *Electronic Design with Off-the-Shelf Integrated Circuits*. West Nyack, New York: Parker Publishing Co., 1980.

Sheingold, Daniel H, ed. *Transducer Interfacing Handbook*. Norwood, MA: Analog Devices, 1980.

Programming

De Jong, Marvin L. *Apple II Assembly Language*. Indianapolis: Howard W. Sams, 1982.

De Jong, Marvin L. *Programming and Interfacing the 6502, with Experiments*. Indianapolis: Howard W. Sams, 1980.

Wagner, Roger. *Assembly Lines: The Book*. North Hollywood, CA: Softalk, 1982.

Laboratory Interfacing

Blankenship, John. *Apple II/IIe Robotic Arm Projects*. Englewood Cliffs, New Jersey: Prentice-Hall, 1985.

Coffron, James W. *The Apple Connection*. Berkeley: SYBEX, 1982.

Titus, Jonathan A., David G. Larsen and C.A. Titus. *Apple Interfacing*. Indianapolis: Howard W. Sams, 1981.

Engelsher, Charles J. *Interfacing and Digital Experiments with Your Apple* . Summit, Pennsylvania: TAB BOOKS, 1984.

De Jong, Marvin L. *Apple II Applications*. Indianapolis: Howard W. Sams, 1983.

Hallgren, Richard C. *Interface Projects for the Apple II*. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1982.

Science

Considine, Douglas M., ed. *Van Nostrand's Scientific Encyclopedia* , 5th Edition. New York: Van Nostrand, 1976.

Weast, Robert C., ed. *Handbook of Chemistry and Physics*, 66th Edition. Boca Raton, FL: The Chemical Rubber Co., 1985.

Appendix D

Software on the *Project Programs* Disk

The files used by each of the projects are listed below. In some cases the same file is used for more than one project. Each file is identified by type and a brief description is provided. The assembly-language, source-code files were all made using the DOS Tool Kit editor and assembler. *Appendix E* is an assembly listing of some of these source-code files.

Project 1. Photogate Timer

BASIC.TIMER	BASIC	simple timing program
PB.TESTER	BASIC	simple PB input-line status test
PHOTOGATE.TIMER	BASIC	main timing program
TIMER	MACHINE	fast timing program (used by PHOTOGATE.TIMER program)
TIMER.SOURCE	TEXT	assembly-language source code used to create the TIMER program
SCHMITT.TIMER	BASIC	main timing program revised for use with Schmitt Trigger circuit
TIMER.S	MACHINE	fast timing program (used by SCHMITT.TIMER program)

Project 2. Reaction Time

PB.TESTER	BASIC	simple PB input-line status test
REACTION.TIMER	BASIC	displays reaction times
TIMER.R	MACHINE	counts reaction time (used by the program REACTION.TIMER)
TIMER.R.SOURCE	TEXT	assembly-language source code used to create the TIMER.R program
R.TIMER.WT.ACC	BASIC	displays reaction times and allows for the use of an accelerator pedal

Project 3. Microphone/Amplifier

FREQUENCY	BASIC	displays the frequency of the input signal
FREQ	MACHINE	counts pulses from the input signal over a 0.5 second period (used by the FREQUENCY program)

Project 4. Humidity Meter

HUMIDITY	BASIC	displays the capacitance and the relative humidity calculated from the frequency of the input signal
FREQ	MACHINE	counts pulses from the input signal over a 0.5 second period (used by the HUMIDITY program)

Project 5. Resistance/Capacitance Meter

METER	BASIC	displays counts on monitor
PDL.CHECK	MACHINE	reads count from PDL input (used by METER program)
PDL.SOURCE.CODE	TEXT	assembly-language source code used to create the PDL.CHECK program

Project 6. IC Temperature Probe

TEMP.SENSOR	BASIC	reads temperature probe
TEMP.CAL	TEXT	calibration data used by the TEMP.SENSOR program
PDL.CHECK	MACHINE	reads count from PDL input (used by TEMP.SENSOR program)
PDL.SOURCE.CODE	TEXT	assembly-language source code used to create the PDL.CHECK program
TEMP.SENSOR.2	BASIC	reads two temperature probes
TEMP.CAL.2	TEXT	calibration data used by the TEMP.SENSOR.2 program

Project 7. Voltage Monitor

VOLTAGE	BASIC	displays the input voltage reading
FREQ	MACHINE	counts pulses from the PB2 input lead over a 0.5 second period (used by the VOLTAGE program)
FREQ.PB0	MACHINE	counts pulses from the PB0 input lead over a 0.5 second period (This program is provided on the disk for those who may want to modify the VOLTAGE program to work with a device that produces a signal on the PB0 lead. See the Teacher's Guide for more information about this modification.)

Project 8. Thermocouple

THERMOCOUPLE	BASIC	displays the temperature difference between the thermocouple junctions
FREQ	MACHINE	counts pulses from the input signal over a 0.5 second period (used by the THERMOCOUPLE program)

Project 9. pH Meter

PH	BASIC	displays the pH of the solution being tested
FREQ	MACHINE	counts pulses from the input signal over a 0.5 second period (used by the PH program)
PH.CAL	BINARY	contains data used to calibrate the pH measurement system

Project 10. Strain Gage

VOLTAGE	BASIC	displays the input voltage reading
STRAIN	BASIC	displays the force applied to the strain gage and allows for calibration of the system
STRAIN.CAL	BINARY	contains data used to calibrate the strain gage
FREQ	MACHINE	counts pulses from the input signal over a 0.5 second period (used by the STRAIN program)

Project 11. Optically-Isolated Switch

SWITCH	BASIC	allows control of each of the annunciator lines
FLASHER	BASIC	turns on each annunciator line briefly in sequence

Project 12. Computer-Controlled Car

MOTOR	BASIC	turns on annunciator lines so that two DC motors may be run forward or backwards

Project 13. Stepper Motor

STEPPER	BASIC	drives a unipolar stepper motor by turning on and off the annunciator lines in the proper order
STEPPER.B	BASIC	drives a bipolar stepper motor by turning on and off the annunciator lines in the proper order

Project 14. A Better Mousetrap

MOUSETRAP	BASIC	controls a better mousetrap

Miscellaneous

HELLO	BASIC	simple "greetings" program
UPDATE	BASIC	displays all known corrections to the book *How to Build a Better Mousetrap*

Appendix E

Assembly- and Machine-Language Program Listings

The assembler listings below were created with the Apple DOS Tool Kit Editor/Assembler:

TIMER
TIMER.R
PDL.CHECK

```
SOURCE FILE: TIMER.SOURCE
0000:            1 **********************
0000:            2 * TIMER.SOURCE
0000:            3 *BY DAVID L. VERNIER
0000:            4 *APR 21, 1986
0000:            5 **********************
----- NEXT OBJECT FILE NAME IS TIMER
8000:            6          ORG  $8000     ;  START AT $8000
8FE0:            7 COUNT.1  EQU  $8FE0     ; LOW BYTE
8FE4:            8 COUNT.2  EQU  $8FE4
8FE8:            9 COUNT.3  EQU  $8FE8
8FEC:           10 COUNT.4  EQU  $8FEC
8FF0:           11 COUNT.5  EQU  $8FF0     ; HIGHEST BYTE
8FFF:           12 ENDMARK  EQU  $8FFF     ;  MARKER TO INDICATE IF IT'S TIME TO ROUTINE
C063:           13 SENSOR   EQU  $C063     ; SW2 ON GAME CONNECTOR
FDF6:           14 COUTZ    EQU  $FDF6     ;  CHARACTER OUTPUT ROUTINE
FD8E:           15 CROUT    EQU  $FD8E     ; CARRIAGE RETURN OUTPUT ROUTINE
C000:           16 KEYBOARD EQU $C000
C010:           17 STROBE   EQU  $C010
0006:           18 STL      EQU  $06       ;  USED TO STALL TO SLOW DOWN TIMING LOOP
8000:           19 **********************
8000:           20 *INITIALIZE ALL COUNTERS AND ENDMARK
8000:A2 00      21 INIT     LDX  #00
8002:A9 00      22 INIT.2   LDA  #00
8004:9D E0 8F   23          STA  COUNT.1,X
8007:E8         24          INX
8008:E0 14      25          CPX  #$14
800A:D0 F6      26          BNE  INIT.2
800C:8D FF 8F   27          STA  ENDMARK   ; SET ENDMARK = 0 (INDICATES NO KEY HAS BEEN PRESSED)
800F:60         28          RTS
8010:           29 **********************
8010:           30 **************************
8010:           31 * STALL
8010:A9 20      32 STALL    LDA  #$20
8012:85 06      33          STA  STL
8014:C6 06      34 LOOP.S   DEC  STL
8016:D0 FC      35          BNE  LOOP.S
8018:60         36          RTS
8019:           37 ***********************
8019:           38 * TIME UNTIL BEAM OFF
8019:A9 AD      39 COUNT.A LDA  #$AD
801B:20 F6 FD   40          JSR  COUTZ     ; PRINT A "-" TO INDICATE COUNTING IS IN PROGRESS
801E:20 10 80   41 CHECK.A JSR  STALL
8021:A9 7F      42          LDA  #$7F      ; FOR COMPARISON WITH SENSOR
8023:D9 63 C0   43          CMP  SENSOR,Y
8026:B0 19      44          BCS  EXIT.A
8028:FE E0 8F   45          INC  COUNT.1,X
802B:D0 F1      46          BNE  CHECK.A
```

```
802D:FE E4 8F           INC  COUNT.2,X
8030:D0 EC              BNE  CHECK.A
8032:FE E8 8F           INC  COUNT.3,X
8035:D0 E7              BNE  CHECK.A
8037:FE EC 8F           INC  COUNT.4,X
803A:D0 E2              BNE  CHECK.A
803C:FE F0 8F           INC  COUNT.5,X
803F:D0 DD              BNE  CHECK.A
8041:60         EXIT.A  RTS
8042:           ***********************
8042:           * TIME UNTIL BEAM ON
8042:A9 DF      COUNT.B LDA  #$DF
8044:20 F6 FD           JSR  COUTZ      : PRINT '_' TO INDICATE COUNTING IS IN PROGRESS
8047:20 10 80   CHECK.B JSR  STALL
804A:A9 7F              LDA  #$7F       : FOR COMPARISON WITH SENSOR
804C:D9 63 C0           CMP  SENSOR,Y
804F:90 19              BCC  EXIT.B
8051:FE E0 8F           INC  COUNT.1,X
8054:D0 F1              BNE  CHECK.B
8056:FE E4 8F           INC  COUNT.2,X
8059:D0 EC              BNE  CHECK.B
805B:FE E8 8F           INC  COUNT.3,X
805E:D0 E7              BNE  CHECK.B
8060:FE EC 8F           INC  COUNT.4,X
8063:D0 E2              BNE  CHECK.B
8065:FE F0 8F           INC  COUNT.5,X
8068:D0 DD              BNE  CHECK.B
806A:60         EXIT.B  RTS
806B:           *************************
806B:           *WAIT TILL BEAM IS BLOCKED
806B:A9 BA      WAIT.A  LDA  #$BA
806D:20 F6 FD           JSR  COUTZ      : PRINT A ':' FOR WAITING FOR BEAM TO BE BLOCKED
8070:A9 A0      CHECK.C LDA  #$A0
8072:CD 00 C0           CMP  KEYBOARD
8075:90 25              BCC  END
8077:A9 7F              LDA  #$7F
8079:D9 63 C0           CMP  SENSOR,Y
807C:90 F2              BCC  CHECK.C
807E:60                 RTS
807F:           *********************************
807F:           *WAIT TILL BEAM IS ON
807F:A9 AE      WAIT.B  LDA  #$AE
8081:20 F6 FD           JSR  COUTZ      : PRINT A '.' TO INDICATE WAITING FOR BEAM TO BE ON
8084:A9 A0      CHECK.D LDA  #$A0
8086:CD 00 C0           CMP  KEYBOARD
8089:90 11              BCC  END
808B:A9 7F              LDA  #$7F
808D:D9 63 C0           CMP  SENSOR,Y
8090:B0 F2              BCS  CHECK.D
8092:60                 RTS
8093:           *********************************
8093:           *PRINT OUT STOP SIGNAL
8093:A9 AA      PRINT.S LDA  #$AA       : ASC '*'
8095:20 F6 FD           JSR  COUTZ
8098:20 8E FD           JSR  CROUT
809B:60                 RTS
809C:           *********************************
809C:           *END ROUTINE
809C:A9 01      END     LDA  #$01
809E:8D FF 8F           STA  ENDMARK    : ENDMARK = 1 INDICATES TIMING IS COMPLETED
80A1:8D 10 C0           STA  STROBE     :  CLEAR KEYBOARD STROBE
80A4:68                 PLA
80A5:68                 PLA             : PULL THE LATEST SUBROUTINE ADDRESS FROM THE STACK
80A6:60                 RTS
80A7:           *****************************
```

```
80A7:           ******************************
80A7:           *  PENDULUM TIMER
80A7:           ******************************
80A7:A2 00      TIMER.C LDX  #$00
80A9:A0 00              LDY  #$0
80AB:20 6B 80           JSR  WAIT.A
80AE:20 42 80           JSR  COUNT.B
80B1:20 19 80           JSR  COUNT.A
80B4:20 42 80           JSR  COUNT.B
80B7:20 19 80           JSR  COUNT.A
80BA:20 93 80           JSR  PRINT.S
80BD:60                 RTS
80BE:           ******************************
80BE:           ******************************
80BE:           *  TRIPLE TIMER
80BE:           ******************************
80BE:A2 00      TIMER.D LDX  #$00
80C0:A0 00              LDY  #$0
80C2:20 6B 80           JSR  WAIT.A
80C5:20 42 80           JSR  COUNT.B
80C8:A2 01              LDX  #$01
80CA:20 19 80           JSR  COUNT.A
80CD:A2 02              LDX  #$02
80CF:20 42 80           JSR  COUNT.B
80D2:20 93 80           JSR  PRINT.S
80D5:60                 RTS
80D6:           ******************************
80D6:           ******************************
80D6:           *  INTERVAL TIMER
80D6:           ******************************
80D6:A2 00      TIMER.E LDX  #$00
80D8:A0 00              LDY  #$0
80DA:20 6B 80           JSR  WAIT.A
80DD:20 42 80           JSR  COUNT.B
80E0:20 93 80           JSR  PRINT.S
80E3:60                 RTS
80E4:           ******************************
80E4:           ******************************
80E4:           * STROBE TIMER
80E4:           ******************************
80E4:A2 00      TIMER.J LDX  #$00
80E6:A0 00              LDY  #$00
80E8:20 7F 80           JSR  WAIT.B
80EB:20 19 80           JSR  COUNT.A
80EE:20 42 80           JSR  COUNT.B
80F1:20 93 80           JSR  PRINT.S
80F4:60                 RTS
80F5:           ******************************
80F5:           ******************************
80F5:           * FLASH TIMER
80F5:           ******************************
80F5:A2 00      TIMER.L LDX  #$00
80F7:A0 00              LDY  #$0
80F9:20 7F 80           JSR  WAIT.B
80FC:20 19 80           JSR  COUNT.A
80FF:20 93 80           JSR  PRINT.S
8102:60                 RTS
8103:           ******************************

*** SUCCESSFUL ASSEMBLY: NO ERRORS
```

```
SOURCE FILE: TIMER.R.SOURCE
0000:           1 * TIMER.R.SOURCE
0000:           2 * VERNIER SOFTWARE
0000:           3 * APRIL 1986
0000:           4 *
----- NEXT OBJECT FILE NAME IS TIMER.R
0300:           5          ORG  $300       ; THIS FILE IS TO BE LOADED AT $300=768
C061:           6 SW0      EQU  $C061      ; SWITCH INPUT #0
03CD:           7 LOW      EQU  $3CD       ; LOW BYTE OF COUNT
03CE:           8 MED      EQU  $3CE       ; MEDIUM BYTE OF COUNT
03CF:           9 HIGH     EQU  $3CF       ; HIGH BYTE OF COUNT
0300:          10 *
0300:A9 00     11 START    LDA  #00
0302:8D CD 03  12          STA  LOW
0305:8D CE 03  13          STA  MED
0308:8D CF 03  14          STA  HIGH       ; STORE ZERO IN ALL COUNTERS
030B:2C 61 C0  15 COUNT    BIT  SW0        ; CHECK SWITCH INPUT SW0
030E:30 0F     16          BMI  EXIT       ; EXIT IF BIT SEVEN IS HICH (PEDAL PRESSED)
0310:EE CD 03  17          INC  LOW
0313:D0 0D     18          BNE  L1
0315:EE CE 03  19          INC  MED
0318:D0 0C     20          BNE  L2
031A:EE CF 03  21          INC  HIGH
031D:D0 0B     22          BNE  LOOP
031F:60        23 EXIT     RTS             ; END OF TIMING ROUTINE
0320:F0 FD     24          BEQ  EXIT       ; EXIT ON OVERFLOW
0322:EA        25 L1       NOP             ; 16 CYCLES OF STALLING START HERE
0323:EA        26          NOP
0324:EA        27          NOP
0325:EA        28          NOP
0326:EA        29 L2       NOP             ; 8 CYCLES OF STALLING START HERE
0327:EA        30          NOP
0328:EA        31          NOP
0329:EA        32          NOP
032A:4C 0B 03  33 LOOP     JMP  COUNT      ; LOOP AGAIN

*** SUCCESSFUL ASSEMBLY: NO ERRORS
```

```
SOURCE FILE: PDL.SOURCE.CODE
0000:              1 ****************************
0000:              2 *PDL.SOURCE.CODE
0000:              3 *BY DAVID L. VERNIER
0000:              4 *VERNIER SOFTWARE
0000:              5 *2920 S.W. 89TH ST.
0000:              6 *PORTLAND. OR 97225
0000:              7 ****************************
----- NEXT OBJECT FILE NAME IS PDL.CHECK
0300:              8          ORG  $300       ; START AT 768 ($300)
C070:              9 PTRIG    EQU  $C070      ; TRIGGER TRANSISTOR
03CE:             10 STOREL   EQU  $3CE       ; LOW BYTE OF COUNTS
03CF:             11 STOREH   EQU  $3CF       ; HIGH BYTE OF COUNTS
C064:             12 PADDLE0  EQU  $C064      ; TO CHECK PDL0
C065:             13 PADDLE1  EQU  $C065      ; TO CHECK PDL1
0300:             14 **********************
0300:             15 ; INITIALIZE 0
0300:A2 60        16 INIT0    LDX  #$60       ; INITIALIZE HIGH BYTE
0302:A0 01        17          LDY  #$01       ; INITIALIZE LOW BYTE
0304:AD 70 C0     18          LDA  PTRIG      ; TURN OFF TRIGGER TRANSISTOR
0307:             19 **********************
0307:2C 64 C0     20 CHECK0   BIT  PADDLE0    ; CHECK FOR CAPACITOR CHARGED
030A:10 31        21          BPL  STORE      ; IF CHARGED. GO STORE RESULTS
030C:C8           22          INY             ; INCREMENT LOW BYTE
030D:D0 06        23          BNE  STALL0     ; NO OVERFLOW. STALL A BIT
030F:E8           24          INX             ; INCREMENT  HIGH BYTE
0310:D0 05        25          BNE  RELOOP0    ; START LOOP OVER
0312:4C 34 03     26          JMP  OVERFLOW   ; OVERFLOW
0315:EA           27 STALL0   NOP             ; NO OPERATION
0316:EA           28          NOP
0317:4C 07 03     29 RELOOP0  JMP  CHECK0
031A:             30 **********************
031A:             31 ; INITIALIZE 1 ; INITIALIZE FOR READING PDL1
031A:A2 60        32 INIT1    LDX  #$60
031C:A0 01        33          LDY  #$01
031E:AD 70 C0     34          LDA  PTRIG      ; TURN OFF TRIGGER TRANSISTOR
0321:             35 **********************
0321:2C 65 C0     36 CHECK1   BIT  PADDLE1    ; CHECK FOR CAPACITOR CHARGED
0324:10 17        37          BPL  STORE
0326:C8           38          INY             ; INCREMENT LOW BYTE
0327:D0 06        39          BNE  STALL1
0329:E8           40          INX             ; INCREMENT  HIGH BYTE
032A:D0 05        41          BNE  RELOOP1
032C:4C 34 03     42          JMP  OVERFLOW   ; OVERFLOW
032F:EA           43 STALL1   NOP
0330:EA           44          NOP
0331:4C 21 03     45 RELOOP1  JMP  CHECK1
0334:             46 **********************
0334:8E CF 03     47 OVERFLOW STX  STOREH     ; STORE HIGH BYTE
0337:8C CE 03     48          STY  STOREL     ; STORE LOW BYTE
033A:4C 54 03     49          JMP  EXIT
033D:             50 ;**********************
033D:             51 ; STORE
033D:8E CF 03     52 STORE    STX  STOREH     ; STORE HIGH BYTE IN MEMORY
0340:8C CE 03     53          STY  STOREL     ; STORE LOW BYTE IN MEMORY
0343:             54 **********************
0343:             55 ; DELAY LOOP
0343:EA           56 DELAY    NOP
0344:EA           57          NOP
0345:EA           58          NOP
0346:C8           59          INY             ; INC LOW BYTE
0347:D0 06        60          BNE  CONTINUE   ; CONTINUE LOOP
0349:E8           61          INX             ; INC HIGH BYTE
034A:D0 05        62          BNE  REPEAT
034C:4C 54 03     63          JMP  EXIT       ; ALL COUNT CYCLE COMPLETED
034F:EA           64 CONTINUE NOP
0350:EA           65          NOP
0351:4C 43 03     66 REPEAT   JMP  DELAY
0354:             67 ********************
0354:60           68 EXIT     RTS

*** SUCCESSFUL ASSEMBLY: NO ERRORS
```

Appendix F

Converting the Programs to ProDOS

The *Project Programs* disk uses the DOS 3.3 operating system. You may want to convert the programs and put them on a ProDOS disk. We have taken some care to write the programs so the converted programs will run properly in ProDOS. To convert the programs, follow this procedure:[1]

1. **Format a blank ProDOS disk.** Use the FILER program on the Apple *ProDOS User's Disk* to format a new disk.[2] If you want to be able to use the new disk as a "start-up disk," also copy the files PRODOS and BASIC.SYSTEM onto it.
2. **Transfer the programs to the ProDOS disk.** Run the program CONVERT, which is on the Apple *ProDOS User's Disk*. Convert and transfer each of the files on the DOS 3.3 *Project Programs* disk to the ProDOS disk. This process takes several minutes.
3. **Try out the ProDOS versions of the programs.** Reboot the converted disk and run a program. The program should function just as the DOS 3.3 version did.

---

[1] This conversion process can also be done using a utility program such as Copy II Plus (version 6.0 or later), Central Point Software, Portland, OR.

[2] If you are unfamiliar with the ProDOS FILER and CONVERT programs, consult your Apple reference manuals.

Appendix G

Teacher's Guide

This section points out some of the possible problems that students may encounter with the projects. It also includes some of our test results and suggestions for ways to use the completed projects in your classroom or lab. We have included a "Feedback Form" in the back of this book. We would like to hear from students and teachers about ingenious project extensions, errors in the book and any other comments you may have.

PROJECT 1: PHOTOGATE TIMER

This project should not be difficult for anyone who has had some experience with soldering. The easiest mistake to make in construction is confusing the leads of the LED and the phototransistor. Also, students sometimes mix up the leads of the cable to the computer. *Appendix A* should help. Careful alignment of the phototransistor and the LED is essential for successful operation. If the student is having trouble with inconsistent operation, check this alignment. If the student follows the troubleshooting procedures and the circuit still does not work, it is likely that parts were overheated and damaged during soldering.

The period of the swing of a pendulum should theoretically be given by the formula below:

$$T = 4\pi^2 (L/g)$$

where:
T = period
L = length of pendulum (center of bob to support point)
g = acceleration due to gravity

Students should be able to precisely verify the relationship between T and L if they are careful about how they measure the length of the pendulum. Students should also be able to verify that the period does not depend on the mass of the bob. This formula is valid only when the pendulum is swinging through a very small angle. The period actually depends to a small extent on the amplitude of the swing. The period increases about 0.5% as the amplitude changes from 0 to 15°. Refer to a college physics text for the series formula used to calculate the period for any amplitude. The variation of the period with amplitude should be noticeable using the photogate timer. This variation may come as a surprise to students who have previously "learned" that period is independent of amplitude.

This project has one of the most extensive "Project Extension" sections. There is a lot of material here for student experimentation.

The photogate the student builds for this project can be a very useful laboratory tool. It is similar to the photogate we recommend for our PRECISION TIMER II program, except that we recommend using a Schmitt Trigger circuit (similar to the one described in the project extensions). The Schmitt trigger circuit eliminates the false triggering problem that occurs with slowly moving objects. With a Schmitt trigger photogate, PRECISION TIMER II can measure times to the nearest 0.1 millisecond. It can time in 14 different timing modes. See *Appendix H* for information on PRECISION TIMER II.

PROJECT 2: REACTION TIME

This is a very simple project in terms of electronics, although it requires some ingenuity and skill on the part of the student to design and construct the brake–pedal apparatus. Some experience with BASIC or other programming languages is recommended.

Most students will enjoy testing themselves and their friends with the brake-pedal simulator. The apparatus may even be of interest to the Driver's Education or Health classes in your school. We purposely did not mention testing people who are under the influence of alcohol or drugs with the project, but such tests might impress students of the dangers of impaired driving.

In our tests, typical reaction times were 0.4 to 0.8 seconds.

Some students may have enough interest in computer programming to want to learn about machine-language programming. The "Supplemental Information" section is intended as an introduction to machine language for those students. They will have to obtain a reference book on machine- and assembly-language to proceed further; recommended books are listed in *Appendix C*.

PROJECT 3: MICROPHONE/AMPLIFIER

Even though there about a dozen components in this project, the instructions are very specific and most students should be able to complete it without too much trouble. The microphone/amplifier can be used for a number of demonstrations involving the physics of music. The range of the microphone is limited by the frequency response of the microphone. It should work to approximately 15,000 hertz. In our tests of the system using labeled tuning forks, the microphone/amplifier has been accurate to within 5 hertz.

The microphone/amplifier built in this project can be used with the Vernier Software program FREQUENCY METER. A cassette recorder with a built-in microphone can also be used for input with the FREQUENCY METER program. See *Appendix H* for more information.

PROJECT 4: HUMIDITY METER

This is a fairly simple project. The circuit uses only five components. The most likely problem would be damaging the Capacitive Humidity Sensor during assembly (especially by overheating during soldering). Note that this project cannot be done on a solderless prototyping board because the capacitance of the prototyping board itself affects the circuit.

The finished humidity meter should be a useful instrument. Physics teachers may want to use it to determine which days are best for static-electricity demonstrations. As mentioned in one of the project extensions, the humidity meter can be combined with IC temperature probes and connected to a single computer, which can be used as the heart of a weather station. Ingenious students may also develop systems for measuring wind direction, wind speed and barometric pressure.

The "dew point" project extension can be used as a laboratory exercise or a demonstration. The project extension that involves modifying the circuit to make a capacitance meter may also be of interest to teachers.

In our tests, the computerized humidity meter agreed with the wet-bulb/dry-bulb relative-humidity readings to within about 5 to 10%. Several of the humidity sensors that we used required recalibration of the HUMIDITY program.

PROJECT 5: RESISTANCE/CAPACITANCE METER

The circuit assembly for this project is fairly easy and there are very few things that can be done wrong. The most likely error would be confusing the leads from the Apple game port or reversing the leads ("+" and "-") on an electrolytic capacitor.

The heart of this project is the charging of a capacitor through a resistor. If a student understands what a capacitor is and how it works, he or she should be able to follow the discussion of how the meter works. Make sure the student reads the section of the *Introduction* that explains how the PDL lines are designed to function. This project would be a good one for a student who knows a little BASIC programming, because there are many possibilities for program development. The program METER on the disk has been kept very primitive to allow a lot of room for improvement.

The completed project provides a method of identifying unknown capacitors or testing questionable capacitors. Our experimental results are as follows:

Our capacitance meter worked very well over a range of 0.01 μF to 10 μF. Values measured agreed with labeled capacitance readings within 10% in all cases checked. Capacitors smaller than about 0.01 μF cannot be read very accurately because their capacitance is always added to the Apple's internal capacitor.

As an ohmmeter, the system worked very well on large resistances, but read low resistance values less accurately. System readings agreed with our digital ohmmeter to within about 1% over the following ranges:

high range (without 10-μF capacitor)	30 kΩ to 2 MΩ
low range (with 10-μF capacitor)	10 kΩ to 50 kΩ

The accuracy was not as good below these ranges. The high range is off by about 10% when measuring 10 kΩ. The low range is off by about 10% when measuring 1 kΩ.

- Homemade capacitors

We made a capacitor with aluminum foil plates that measured 18 cm by 25 cm. We got the following capacitance readings with various insulators used between the sheets of foil:

Saran Wrap	.0059 μF
waxed paper	.0044 μF
1 layer plastic page protector	.0061 μF
2 layers page protector	.0017 μF

Notice that these capacitances are very small—barely measurable by this instrument. A capacitor made in a roll consisting of two aluminum sheets measuring 22 cm by 4 m, and separated by sheets of Saran Wrap had a capacitance of 0.067 μF. There is quite a difference in the dielectric constant of different plastic materials. Students may be able to find a material that comes in very thin sheets and has a high dielectric constant. There are a lot of opportunities for experimentation here. If the thickness of the insulating material can be measured accurately, the dielectric constant of various materials can even be measured and compared with values listed in physics texts. The relationship between capacitance, plate area and plate separation can also be studied.

- Temperature stability experiments

Many capacitors and resistors have large temperature coefficients. Here are some experimental results:

Device	Increase in Temperature	% Change in Value
ceramic capacitor	20°C	-3%
electrolytic capacitor	20°C	+1.7%
tantalum capacitor	20°C	+0.7%
mylar capacitor	20°C	-0.02%
metalized-polyester capacitor	20°C	+0.05%
carbon-composition resistor	60°C	-1.5%
metal-film resistors	60°C	-0.2%

Notice that in general, capacitors increased in value and resistors decreased in value as they were heated. Metal-film resistors and "plastic-insulator" capacitors (mylar, polyester, etc.) are remarkably stable, which is why they are recommended for many of the projects in this book.

- Conductivity of soil experiments

These experiments work well with this system because the resistance can be very high. The resistance of the soil varies dramatically as the moisture content increases. Typical resistance readings with probes a few centimeters apart in the soil:

dry soil	several million ohms
wet soil	several hundred-thousand ohms

PROJECT 6: IC TEMPERATURE PROBE

The circuit for this project is easy to build and the finished probe is a handy, general purpose (-55°C to +150°C) thermometer for lab use or for use in weather studies. If some of the program improvements are done, the IC temperature probe will do some things that a normal thermometer will not do, such as automatically taking readings at predetermined intervals or creating graphs. The temperature probe should be accurate to within 0.5°C if it is carefully calibrated. If calibrated over a small temperature range, the accuracy should be better. The probe will not read correctly if it gets wet inside. If it gets wet, dry it out and reseal it with silicone sealant.

Vernier Software sells a program called TEMPERATURE PLOTTER that allows the use of up to four of these IC temperature probes. This program collects time and temperature data, saves data in memory and on disk and graphs temperature versus time. It incorporates almost all of the program improvements suggested as project extensions. Vernier Software also sells assembled temperature-probe systems. See *Appendix H* for more information.

PROJECT 7: VOLTAGE MONITOR

This is a key project since it is used in three other projects: *Thermocouple*, *pH Meter* and *Strain Gage*. The Voltage Monitor circuit involves quite a few components; therefore there are many places for students to make mistakes. Encourage the students to take their time and double-check every connection before plugging in the circuit. If the students use the recommended circuit board, the exact placement of the components can be checked easily by comparing the circuit board with the diagrams in the "Step-by-Step Instructions."

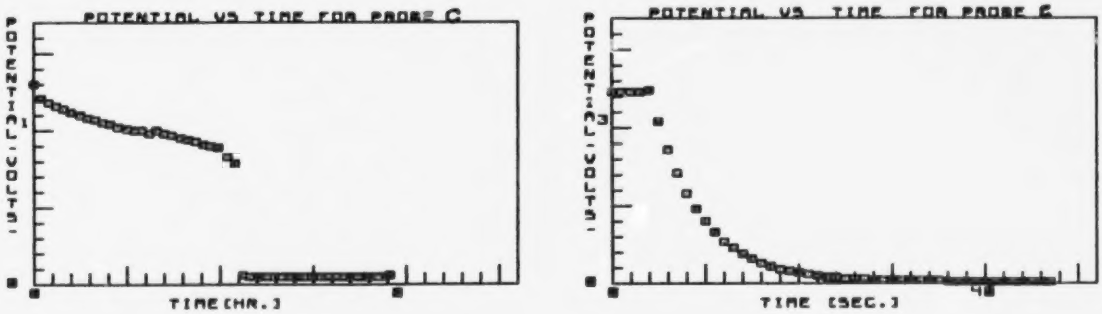
Students should be warned not to connect the Voltage Monitor across potential differences larger than 3.5 volts. The unit will not be damaged by voltages up to 5 volts, but it will erroneously read zero volts. Also, remind students to observe the proper polarity for the test leads. The Voltage Monitor circuit can be damaged if potentials well outside the recommended range are used. Adding the zener-diode, input-protection circuit (as explained in one of the project extensions) minimizes this potential danger. Even with the zener-diode circuit, the Voltage Monitor and possibly the computer itself could be damaged if potential differences larger than 20 volts are used.

Two things to keep in mind when you use the Voltage Monitor are: (1) The Voltage Monitor does what is called *single-ended* voltage measurement; that is, all voltages measured are compared to the ground of the Apple's power supply. (2) The Voltage Monitor cannot accurately read rapidly changing voltage signals, since it counts pulses over a one-half second period to determine the voltage. In effect, the voltage is averaged over the half-second sample period.

The accuracy of the Voltage Monitor circuit is very good. In tests of typical units over the entire 0 to 3.5 volt range, the maximum error (assuming our digital voltmeter was perfect) was 0.02 volts. The effective input impedance of the Voltage Monitor is several million ohms. If the zener-diode protection circuit is added, the effective input impedance is reduced somewhat due to leakage of the zener diode.

The Voltage Monitor can be used for many labs and demonstrations. It is essentially the same as the "16-pin Voltage Input Unit" sold by Vernier Software for use with the program VOLTAGE PLOTTER. The VOLTAGE PLOTTER program collects, displays and graphs voltage data.

The two graphs below were made using a Voltage Monitor and the VOLTAGE PLOTTER program. They show results of the two "Additional Experiments" suggested in the Project Extensions for this project. The left graph shows the discharging of an alkaline battery. The right graph shows a capacitor discharge curve.



SAMPLE GRAPHS MADE WITH A VOLTAGE MONITOR AND VOLTAGE PLOTTER

Note to teachers who use 9-pin Voltage Input Units from Vernier Software:

> A program named FREQ.PB0 is included on the *Project Programs* disk to allow you to use a 9-pin Voltage Input Unit (VIU) from Vernier Software in place of the Voltage Monitor. A 9-pin VIU produces a signal just like the Voltage Monitor, except it uses the PB0 line instead of the PB2 line. To work with a 9-pin VIU, the BASIC programs on the *Project Programs* disk that normally use the FREQ program will need to be modified so that they load and use the FREQ.PB0 program. To modify the VOLTAGE program, for example, you need to change line 160 to read:

```
160  PRINT D$; "BLOAD FREQ.PB0,A$8300 :  REM LOAD MACHINE CODE TO
READ PB0 INPUT
```

> A similar change can be made on the programs THERMOCOUPLE, PH, and STRAIN. Making these changes will allow you or your students to do the projects *Thermocouple*, *pH Meter* and *Strain Gage* using the 9-pin VIU.

PROJECT 8: THERMOCOUPLE

The thermocouple circuit should not be a difficult project for a student who has built other electronic circuits. However, the student will need to read a simple schematic diagram in order to lay out the circuit. The most confusing component for the student is the potentiometer. The finished project is useful for several demonstrations. One example is to use the thermocouple to show students that the temperature at the tip of a bunsen-burner flame is much hotter than the temperature low in the flame. You are encouraged to write for the free literature on temperature measurement from Omega Industries. The address is included in the project.

The TLC271 op amp was selected for this project for two reasons: (1) It is specifically designed for use with a single power supply (no "-" power supply lead); and (2) It was available at Radio Shack stores and in their 1986 catalog. Unfortunately, this op amp is not in the 1987 Radio Shack catalog. As a result, the student may have some difficulty obtaining it. If necessary, the project can be done with a different "high input impedance" op amp. The offset circuit shown in the schematic diagrams for this project will probably need to be changed for a different op amp; consult the manufacturer's data sheets.

This thermocouple temperature-measurement system is not as accurate as the IC Temperature Probe (Project 6). The thermocouple circuit will typically be accurate to within 5 or 10 degrees Celsius.[1] However, it is useful over a very wide range: 0 to 1350°C. It can even be used to measure temperatures below 0°C if the reference and probe junctions are reversed.[2]

This thermocouple circuit is similar to one described in the manual for the Vernier Software VOLTAGE PLOTTER program.

PROJECT 9: pH METER

In terms of electronics, this project is not especially complex. However, the exact placement of the components on the circuit board is not specified. As a result, the student has to make a few more decisions than on some projects. A factor that may make this project difficult is the pH electrode. We are hoping most students will be able to obtain a pH electrode from an old pH meter around their school and avoid the expense of purchasing a new electrode. The problem with this is that old electrodes can be cantankerous. If the electrode has not been stored properly, it may take a while for it to "come back to life." The manufacturers usually recommend soaking the electrode in a particular solution. If possible, refer to the information sheet from the electrode manufacturer.

We have had very good results with this pH measurement system. In our tests, the pH value displayed by the program is typically within 0.05 of the labeled value of the buffer. We use nickel-cadmium batteries for the power supply because they do not change voltage as much as conventional batteries as they lose their charge. This minimizes the drift in calibration that may occur as the batteries run down.

If you do not have commercial buffer solutions available, "recipes" are available for making buffers from common laboratory chemicals. For example, see *An Introduction to Practical Biochemistry* by David T. Plummer, McGraw Hill, London (1978) pp. 350-351.

The distilled water experiment suggested in the "Project Extensions" involves the formation of carbonic acid by the absorption of carbon dioxide from the air. The pH of the distilled water should decrease.

This pH amplifier circuit is similar to one described in the manual for the Vernier Software VOLTAGE PLOTTER program.

PROJECT 10: STRAIN GAGE

This is one of the most difficult projects in the book. The strain gages are small and fragile and some students will have a difficult time gluing and soldering leads to them. The bridge-amplifier circuit is somewhat complex. In addition, the problem of interference can be frustrating. If the student does a sloppy job in soldering together the strain gages, bridge circuit and amplifier, the result will be a noisy output signal. This is truly a project in which neatness counts. In some of our early prototypes, the output voltage would increase by half a volt whenever a hand was held near the strain gages.

Do not expect to have this project take the place of an electronic balance. The core project should be accurate to within 5%. The accuracy is improved somewhat if the suggestions for minimizing interference are followed (including housing the circuit in a metal box). Using a commercial instrumentation amplifier, as shown in one of the project extensions, will also improve accuracy and stability. The use of a voltage regulator on the excitation voltage is another step that significantly improves the performance of the system.

---

[1] Unless you have a precisely controlled oven or high temperature thermometers, checking the calibration of the thermocouple system is not easy. You might try verifying the melting points of some metals, such as aluminum and copper.

[2] The linearity of the thermocouple voltage versus temperature difference is not as good at very low temperatures, so the accuracy will decrease.

This project can start the student on a number of interesting projects using strain gages. Several ideas are listed in the "Project Extensions" and others are included in the free strain gage literature available from the manufacturers. Addresses are included in the project.

This type of force measurement has been used successfully in the Hillsboro High School District (Hillsboro, Oregon) Bridge Building Contest for several years. The force is applied to the model bridges with a screw jack. A modified version of the STRAIN program monitors the force and also "remembers" the maximum force applied to a particular bridge. For bridge testing, it is very helpful to have the program display force readings in large low-resolution graphics characters so the whole room can see them. Writing such a program is a good student project.

This strain-gage circuit is similar to one described in the manual for the Vernier Software VOLTAGE PLOTTER program.

PROJECT 11: OPTICALLY-ISOLATED SWITCHES

This is a complex circuit that should be attempted only by students with some circuit-assembly experience. The circuit was designed very conservatively to protect the computer and to allow for controlling a wide variety of external circuits.

Note that this switch cannot be used with AC circuits. It is relatively easy to build an AC switch circuit that can be controlled by the annunciator lines. The easiest way is to use solid-state relays. We chose not to include such circuits in this book because of the potential danger to students. Several of the reference books listed in *Appendix C* discuss controlling AC electricity through the game port. An article by Steve Ciarcia in the December 1984 issue of *Byte* discusses several circuits to control both DC and AC with a computer.

It is possible to build just one or two of the Optically-Isolated Switch circuits if that is all that is needed. Physics teachers may be interested in the project extension that involves building a "monkey gun." It requires only one Optically-Isolated Switch.

Notice that this project cannot be connected to the 9-pin external game port on an Apple IIe.

Three of the other projects (*Computer-Controlled Car, Stepper Motor, and A Better Mousetrap*) build on this project.

PROJECT 12: COMPUTER-CONTROLLED CAR

This project is intended as a simple introduction to robotics. Students looking for a science project often think first of a project involving a robot. This project (and/or the next one, *Stepper Motor*) can be a good start for experiments of this kind. There are a lot of possibilities for project extensions. This project is designed around the Dual Drive Gear Motors Unit, which is sold as a surplus part from toy manufacturing. If the student uses this unit, the project is fairly straightforward. The project can be done with any two 5-volt DC motors, but it will be more difficult.

PROJECT 13: STEPPER MOTOR

If the right stepper motor is used, this is an easy project. A unipolar or four-phase stepper motor is needed for the core project. Bipolar (2-phase) stepper motors can be used in a more complex circuit described in the project extensions. A stepper motor with leads that are not identified may be frustrating to the student. If it is an Airpax motor (the type used in the Vernier Software parts kit), the information in the "Step-by-Step Instructions" should provide all of the information the student needs.

Most students will find stepper motors interesting. This project and the previous one (*Computer-Controlled Car*) are intended as an introduction to robotics. There are probably dozens of uses for stepper motors in science classes; examples include controlling automatic animal feeders, scanning across a diffraction pattern and aiming solar panels.

PROJECT 14: A BETTER MOUSETRAP

This project is only partially serious. We have included it as an example of putting together some of the ideas in the book for a "practical" device. We actually used this device to catch fourteen mice in our garage. We let them all go outside in a field, so we don't know for sure that we didn't catch the same mouse fourteen times. We had a lot of fun doing it and we learned some things at the same time. We hope that a few students do the same with this or one of the other projects in this book.

Appendix H

Other Vernier Software Products

Vernier Software has been publishing science software for the Apple computer since 1981. The following is a list of software and hardware available. If you are interested in receiving a complete catalog of Vernier Software products, write or call:

Vernier Software
2920 S.W. 89th Avenue
Portland, OR 97225
(503) 297-5317

◊ **VOLTAGE PLOTTER**: Allows the Apple II+/IIe/IIc to be used as a general purpose voltmeter, chart recorder or oscilloscope. The accompanying manual describes additional circuits that allow monitoring of pH, force (with strain gages) and temperature (with thermocouples). Includes a 68-page manual. $39.95

VOLTAGE PLOTTER requires one of the following hardware devices:

- Voltage Monitor: Project 7 of this book
- Voltage Input Unit: Similar to a Voltage Monitor, except built on a custom integrated-circuit board and including zener diode input protection. $35.00 assembled or $25.00 parts kit
- Advanced Interfacing Board: A 12-bit, 8-input analog-to-digital converter board that plugs into one of the slots inside an Apple II+ or IIe. $225.00

◊ **PRECISION TIMER II**: Turns the Apple into a laboratory timer. Measures times to 0.1 millisec in 14 different timing modes. Time data is collected and the computer can help analyze it. Times may be displayed in large block digits, saved on disk or graphed. Used with PASCO scientific photogates or with homemade photogates similar to the photogate described in Project 1 of this book. Complete instructions for building a 3-photogate system with a Schmitt trigger are included in the program's 54-page manual. $39.95

- Apple II+/IIe 3-photogate parts kit $40.00
- Apple IIc 2-photogate parts kit $35.00

◊ **TEMPERATURE PLOTTER**: Allows the Apple II+/IIe/IIc to monitor temperatures in the range -55 to +150°C. Temperature data may be stored on disk, edited, and graphed. Temperature units, delay between readings and type of clock used are under program control. Includes a 55-page manual. $39.95

The program requires a temperature-probe system that uses the same sensor as the IC Temperature Probe (Project 6 of this book).

	parts kit	assembled
• 2-probe system	$20.00	$40.00
• 4-probe system	$30.00	$60.00

◊ **FREQUENCY METER**: Allows the Apple II+/IIe/IIc computer to be used as a general purpose audio-range frequency meter in the laboratory. Frequency readings may be displayed in large digits or saved in a data table in memory. The data table may be edited, saved on disk or graphed. Several special options are included to help demonstrate the physics of music. Includes a 44-page manual and a fife (a simple musical instrument). $39.95

FREQUENCY METER can be used with a microphone and tape recorder connected to the Apple's cassette input port. No other hardware is required. Alternatively, the program can be used with a microphone/amplifier circuit like the one described in Project 3 of this book. A parts kit for the microphone/amplifier circuit is available for $12.00.

◊ **GRAPHICAL ANALYSIS III**: Plots well-labeled graphs of experimental data. After data is entered, modified versions of the graph (e.g., with the x-axis data squared before plotting) may be quickly drawn to help in the search for the relationship between the variables. Semi-log and log-log graphs can be made. The linear regression "best fit" line can be included on the graph. Data tables may be edited, printed, or saved on diskette. Copies of the graphs may be made on a printer. Compatible with AppleWorks. $24.95

◊ **RAY TRACER**: Uses Apple high-resolution graphics to illustrate the principles of geometrical optics. Reflection, refraction, Snell's Law, total internal reflection, image formation with lenses and mirrors, dispersion, and spherical and chromatic aberration may all be demonstrated. May be used as a lecture demonstration aid or as an educational game. Includes a 19-page manual. $24.95

◊ **ORBIT II**: Simulates the motion of an earth satellite. Students are presented with a series of challenges. They control launch angle and speed and may use thruster rockets to alter the orbital path in the attempt to accomplish the task. Includes a 23-page manual. $24.95

◊ **CHARGED PARTICLES II**: Simulates the motion of various charged particles in magnetic and electric fields. A series of challenges is presented for student solution. Many important concepts of electromagnetism may be explored. The 27-page manual includes student worksheets. $24.95

◊ **PROJECTILES II**: Allows students to experiment with projectile motion, including the effects of air resistance and wind. Launch may be from ground level or from the top of a cliff. Moving targets are sometimes used to make the calculation of the correct inputs more difficult. Includes a 21–page manual. $24.95

◊ **WAVE ADDITION II**: Graphically demonstrates the superposition of waves. Various phenomena such as interference, beats, the "shaping" of waves by the addition of harmonics and Fourier synthesis can be demonstrated. Includes an 18-page manual. $24.95

◊ **VECTOR ADDITION II**: Graphically demonstrates the head-to-tail addition of vectors. Magnitudes and directions are input; the vectors and their resultant are drawn. Numerical values of the resultant and the components are also calculated. Includes a 13-page manual. $24.95

◊ **KINEMATICS II**: Students control the starting speed, acceleration and elapsed time for the motion of a truck that moves across the screen. They attempt to cause the truck to move according to preassigned parameters. Student results may be saved in a file for teacher use. Includes an 18-page manual. $24.95

Parts Kit Order Form

Vernier Software produces parts kits for several of the projects in this book. You may find these parts kits convenient because some of the components for these projects are difficult to obtain in single quantities. Each kit contains all of the necessary components (listed in the parts list at the beginning of the project). Parts kits that connect directly to the computer will include a 16-pin game-port cable. The price for each kit is listed below.[1] Add $3.00 per order (not per kit) for shipping and handling.

Please enclose either school purchase order or check. Sorry, we do not accept credit card orders.

Project	Parts Kit Price	Quantity	Amount
Photogate Timer	$8.00	______	______
Microphone/Amplifier	$12.00	______	______
Humidity Meter	$23.00	______	______
IC Temperature Probe	$13.00	______	______
Voltage Monitor	$24.00	______	______
Thermocouple	$13.00	______	______
Strain Gage	$30.00	______	______
Optically-Isolated Switches	$30.00	______	______
Stepper Motor	$8.00	______	______
		Shipping	**+$3.00**
		Total-->	______

☐ Please send ___ copies of *How to Build a Better Mousetrap* @ $24.95 (includes shipping)

☐ Please send me a catalog of Vernier Software products.

Ship to:

Name____

School_______________________________________

Address______________________________________

City, State, Zip_________________________________

---

[1] Prices and availability of these parts kits are subject to change.

Feedback Form

We would like to hear from you. Please contact us if you have comments or suggestions about this book.

Suggestions for improving the book (corrections, clarifications, etc.):

Interesting project extensions you have developed:

Other comments:

Name ____

School ____

Address ____

City, State, Zip __________________________________

I am a: ☐ Student ☐ Teacher

Send to:

Vernier Software
2920 S.W. 89th St.
Portland, Oregon 97225

HOW TO BUILD
A BETTER MOUSETRAP
and 13 Other Science Projects Using the Apple II®

by David L. Vernier

If you have an Apple II+ or Apple IIe computer and if you are interested in science, then this book is for you. Included are 14 science projects for students (or teachers) who want to learn about laboratory interfacing via the Apple II+/IIe game port. The projects are aimed at students who have minimal experience with computers or electronics. Step-by-step instructions are provided for the assembly of the electronic circuits. A disk of sample programs is included to allow you to try out each project. This is a great book for independent study or science fair projects!

Projects:

- Photogate Timer
- Reaction Time
- Microphone/Amplifier
- Humidity Meter
- Resistance/Capacitance Meter
- IC Temperature Probe
- Voltage Monitor
- Thermocouple
- pH Meter
- Strain Gage
- Optically-Isolated Switches
- Computer-Controlled Car
- Stepper Motor
- A Better Mousetrap

About the Author:

David L. Vernier is a former high school physics teacher of fourteen years. He currently writes and publishes science software for high schools and colleges.

Published by:

Vernier Software
2920 S.W. 89th Street
Portland, Oregon 97225
(503) 297-5317

Price: $24.95

ISBN 0-918731-16-X